

NEIMAR NEITZEL

**PROTOCOLO DE TRANSPARÊNCIA DA MOBILIDADE  
PARA REDES AD HOC MÓVEIS**

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre. Programa de  
Pós-Graduação em Informática, Setor de Ciências  
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Carlos Pessoa Albini

CURITIBA

2013

NEIMAR NEITZEL

**PROTOCOLO DE TRANSPARÊNCIA DA MOBILIDADE  
PARA REDES AD HOC MÓVEIS**

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre. Programa de  
Pós-Graduação em Informática, Setor de Ciências  
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Carlos Pessoa Albini

CURITIBA

2013

Neitzel, Neimar

Protocolo de transparência da mobilidade para redes ad hoc móveis / Neimar Neitzel. – Curitiba, 2013.

81 f.: il., tab.

Dissertação (mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2013.

Orientador: Luiz Carlos Pessoa Albini

1. Redes de Computadores – Protocolos. 2. Sistemas de Comunicação móvel. I. Universidade Federal do Paraná. II. Albini, Luiz Carlos Pessoa. III. Título.

CDD 004.62



Ministério da Educação  
Universidade Federal do Paraná  
Programa de Pós-Graduação em Informática

## PARECER

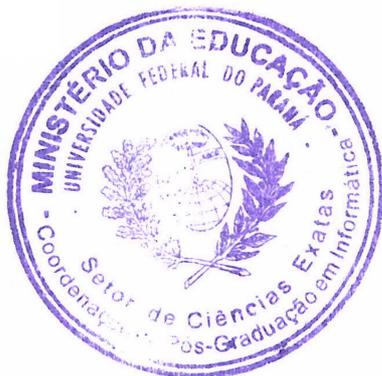
Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno Neimar Neitzel, avaliamos o trabalho intitulado, "*Protocolo de Transparência da Mobilidade para Redes Ad Hoc Móveis*", cuja defesa foi realizada no dia 26 de fevereiro de 2013, às 15:30 horas, no Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato.

Curitiba, 26 de fevereiro de 2013.

Prof. Dr. Luiz Carlos Pessoa Albini  
DINF/UFPR – Orientador

Prof. Dr. Alcides Calsavara  
PUC/PR – Membro Externo

Prof. Dr. Roberto Hexsel  
DINF/UFPR – Membro Interno



## AGRADECIMENTOS

Agradeço primeiramente a Deus que me deu saúde para terminar. A Rafaella que sempre esteve junto nos momentos alegres e principalmente nos difíceis. Agradeço também aos meus pais, irmãs e a tia Dolores que sempre me apoiaram e incentivaram a continuar em frente.

Agradeço muito ao meu orientador Albini, por todo aprendizado e tempo despendido comigo. Todo conhecimento passado em sala, reuniões e conversas de corredor serão levados comigo e com certeza vou querer passar adiante. Pela paciência e compreensão nos momentos de necessidade e ainda por cobrar quando foi preciso e me orientar durante todo esse tempo.

Aos amigos que fiz nessa jornada, nos momentos de festa, bolos, paçocas, cafés, almoços e escaladas. Entre eles não posso deixar de citar Urlan, Elisa, Fernando, Marco, Luiz Fernando, Cinara, Leila, Renan, Eduardo, entre outros. Aos amigos de outras datas que estão sempre apoiando e auxiliando nos momentos difíceis, (Willian e Dabna, Marcos, Andres, Andre, Newton, Joylan, entre outros). Aos membros do LABI que me deram a base sólida durante a graduação e sempre me incentivaram a seguir nessa carreira, meu obrigado Renato, Wu e Huei, agradecendo a vocês agradeço aos demais membros.

Finalmente agradecer ao grupo NR2 onde conheci grandes pesquisadores e amigos. Obrigado pela infra-estrutura que foi crucial, principalmente no final do trabalho e pelos momentos de estudo e descontração.

## RESUMO

As redes Ad Hoc Móveis (*Mobile Ad Hoc Networks* (MANETs)) são redes sem fio e sem uma infra-estrutura pré-existente. Elas possuem uma administração centralizada, sendo mantidas de modo autônomo e distribuído. Os nós podem se mover livremente pelo ambiente, criando uma topologia de rede dinâmica, gerenciada e mantida por protocolos de roteamento distribuídos. As MANETs podem ser utilizadas em diferentes áreas, entre elas, cita-se as redes espontâneas e como uma extensão da Internet, possibilitando maior mobilidade aos usuários. Uma das características das MANETs é a mobilidade dos nós. O que, em alguns casos, acaba sendo um problema, pois a mobilidade degrada o desempenho do protocolo TCP e das aplicações utilizadas na Internet. Para que as aplicações e o TCP possam funcionar adequadamente nas MANETs são necessárias alterações nas implementações deles, o que demanda tempo e acarreta em custos adicionais no seu desenvolvimento e manutenção. Para possibilitar que as aplicações e os protocolos existentes possam ser utilizados sem alterações nas MANETs, propõe-se neste trabalho um protocolo capaz de deixar a mobilidade transparente, por meio de uma rede virtual fixa. O sistema proposto é denominado Protocolo Virtual de Transparência da Mobilidade (*Virtual Transparent Mobility Protocol* (VTMP)). O VTMP atua entre os protocolos de roteamento e de transporte, sem alterar a implementação de nenhum deles, recebendo os pacotes da camada de transporte e encaminhando-os por meio da rede virtual fixa, criada sobre a rede real móvel. O objetivo do VTMP é permitir que o TCP possa funcionar sobre a rede móvel sem qualquer alteração, possibilitando que qualquer aplicação tradicional que use o TCP também possa ser utilizada sem alterações. Neste trabalho, o VTMP foi implementado e avaliado em diferentes ambientes e condições de tráfego.

## ABSTRACT

Mobile Ad Hoc Networks (MANETs) are wireless networks without any preexisting infrastructure. They have a centralized administration and are maintained in an anonymous and distributed way. Nodes can move freely throughout the environment, creating a dynamic network topology managed and maintained by distributed routing protocols. MANETs can be used in different areas, including spontaneous networks and Internet extension, allowing the users to move while connected to the Internet. One of the main characteristics of MANETs is the mobility of the nodes. In some cases, mobility becomes a problem, since it deteriorates the TCP protocol and the performance of the Internet applications. For the correct functioning of applications and the TCP protocol on MANETs, some changes are mandatory their implementations, which takes time and entails additional costs in their development and maintenance. In order to use existing protocols and applications on MANETs without implementation changes, this work proposes a protocol capable of abstracting the mobility through a fixed virtual network. The proposed system is entitled Virtual Transparent Mobility Protocol (VTMP). The VTMP acts between the routing and transport protocols, without changing their implementation, receiving packets from the transport layer and forwarding them through the fixed virtual network, created upon the real mobile network. The goal of VTMP is to allow TCP to operate on mobile network without any change, enabling applications that uses the traditional TCP to be used without further changes. In this work, VTMP has been implemented and analyzed in different environments and traffic conditions.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Problema e Objetivo . . . . .	2
1.2	Estrutura do Trabalho . . . . .	5
<b>2</b>	<b>FUNDAMENTOS</b>	<b>6</b>
2.1	Protocolos de Roteamento . . . . .	6
2.1.1	<i>Ad Hoc On-Demand Distance Vector</i> . . . . .	7
2.2	Protocolo de Transporte . . . . .	8
2.2.1	TCP Tahoe . . . . .	9
2.2.2	TCP Reno . . . . .	11
2.2.3	TCP Vegas . . . . .	12
2.2.4	TCP New Jersey . . . . .	13
2.3	Considerações Finais . . . . .	14
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>15</b>
3.1	<i>Cross-layer Congestion Control for TCP (C<sup>3</sup>TCP)</i> . . . . .	15
3.1.1	<i>TCP with Gateway Adaptive Pacing (TCP-GAP)</i> . . . . .	16
3.1.2	TCP Vegas MANET (TCP Vegas_M) . . . . .	17
3.2	<i>TCP With One Way (TCP-WOW)</i> . . . . .	17
3.3	ADHOCTCP . . . . .	18
3.4	Considerações Finais . . . . .	18
<b>4</b>	<b>VTMP – PROTOCOLO VIRTUAL DE TRANSPARÊNCIA DA MO- BILIDADE</b>	<b>20</b>
4.1	Estrutura Virtual . . . . .	22
4.2	Descrição do VTMP . . . . .	23
4.2.1	Pacotes . . . . .	24

4.2.2	Atualização e Manutenção das Rotas . . . . .	25
4.2.3	Roteamento de Pacotes . . . . .	26
4.2.4	Otimizações . . . . .	26
<b>5</b>	<b>DEFINIÇÃO DAS SIMULAÇÕES</b>	<b>28</b>
5.1	Métricas Utilizadas . . . . .	28
5.2	Parâmetros de Simulação . . . . .	29
5.3	Definição do Tempo de Atualização de Rotas do VTMP . . . . .	31
5.4	Considerações Finais . . . . .	41
<b>6</b>	<b>RESULTADOS DA AVALIAÇÃO EXPERIMENTAL DO PROTO- COLO VTMP</b>	<b>42</b>
6.1	Resultados do protocolo VTMP sob o TCP Reno . . . . .	43
6.2	Avaliação experimental do protocolo VTMP sob o TCP Tahoe . . . . .	53
6.3	Resultados do protocolo VTMP sob o TCP Vegas . . . . .	59
6.4	Resultados do protocolo VTMP sob o TCP New Jersey . . . . .	65
6.5	Considerações Finais . . . . .	71
<b>7</b>	<b>AVALIAÇÃO EXPERIMENTAL DO FUNCIONAMENTO DO VTMP SOB UMA APLICAÇÃO REAL</b>	<b>73</b>
<b>8</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>75</b>
	<b>BIBLIOGRAFIA</b>	<b>81</b>

## LISTA DE FIGURAS

1.1	Pilha de protocolos modelo TCP/IP. . . . .	2
2.1	Janela de congestionamento TCP Tahoe [17]. . . . .	10
2.2	Janela de congestionamento TCP Reno [17]. . . . .	11
4.1	Posicionamento do Protocolo VTMP. . . . .	20
4.2	Rede virtual vs Rede real. . . . .	21
4.3	Exemplo de Hipercubo. . . . .	23
4.4	Pacote do VTMP. . . . .	25
5.1	Atraso considerando arquivos de 100 KB. . . . .	32
5.2	Atraso considerando arquivos de 800 KB. . . . .	32
5.3	Taxa de Entrega considerando arquivos de 100 KB. . . . .	34
5.4	Taxa de Entrega considerando arquivos de 800 KB. . . . .	34
5.5	Vazão Efetiva considerando arquivos de 100 KB. . . . .	35
5.6	Vazão Efetiva considerando arquivos de 800 KB. . . . .	36
5.7	Descarte de pacotes considerando arquivos de 100 KB. . . . .	37
5.8	Descarte de pacotes considerando arquivos de 800 KB. . . . .	37
5.9	Descarte de pacotes por falta de rota considerando arquivos de 100 KB. . . . .	38
5.10	Descarte de pacotes por falta de rota considerando arquivos de 800 KB. . . . .	39
5.11	Sobrecarga de pacotes de controle considerando arquivos de 100 KB. . . . .	39
5.12	Sobrecarga de pacotes de controle considerando arquivos de 800 KB. . . . .	40
6.1	Atraso considerando arquivos de 100 KB. . . . .	43
6.2	Atraso considerando arquivos de 800 KB. . . . .	44
6.3	Taxa de Entrega considerando arquivos de 100 KB. . . . .	45
6.4	Taxa de Entrega considerando arquivos de 800 KB. . . . .	46
6.5	Vazão Efetiva considerando arquivos de 100 KB. . . . .	46

6.6	Vazão Efetiva considerando arquivos de 800 KB. . . . .	47
6.7	Descarte de pacotes considerando arquivos de 100 KB. . . . .	48
6.8	Descarte de pacotes considerando arquivos de 800 KB. . . . .	49
6.9	Descarte de pacotes por falta de rota considerando arquivos de 100 KB. . .	49
6.10	Descarte de pacotes por falta de rota considerando arquivos de 800 KB. . .	50
6.11	Sobrecarga considerando arquivos de 100 KB. . . . .	51
6.12	Sobrecarga considerando arquivos de 800 KB. . . . .	52
6.13	Atraso considerando arquivos de 100 KB. . . . .	53
6.14	Atraso considerando arquivos de 800 KB. . . . .	54
6.15	Taxa de Entrega considerando arquivos de 100 KB. . . . .	54
6.16	Taxa de Entrega considerando arquivos de 800 KB. . . . .	55
6.17	Vazão Efetiva considerando arquivos de 100 KB. . . . .	55
6.18	Vazão Efetiva considerando arquivos de 800 KB. . . . .	56
6.19	Descarte de pacotes considerando arquivos de 100 KB. . . . .	56
6.20	Descarte de pacotes considerando arquivos de 800 KB. . . . .	57
6.21	Descarte de pacotes por falta de rota considerando arquivos de 100 KB. . .	57
6.22	Descarte de pacotes por falta de rota considerando arquivos de 800 KB. . .	57
6.23	Sobrecarga considerando arquivos de 100 KB. . . . .	58
6.24	Sobrecarga considerando arquivos de 800 KB. . . . .	58
6.25	Atraso considerando arquivos de 100 KB. . . . .	59
6.26	Atraso considerando arquivos de 800 KB. . . . .	60
6.27	Taxa de Entrega considerando arquivos de 100 KB. . . . .	60
6.28	Taxa de Entrega considerando arquivos de 800 KB. . . . .	61
6.29	Vazão Efetiva considerando arquivos de 100 KB. . . . .	61
6.30	Vazão Efetiva considerando arquivos de 800 KB. . . . .	62
6.31	Descarte de pacotes considerando arquivos de 100 KB. . . . .	62
6.32	Descarte de pacotes considerando arquivos de 800 KB. . . . .	63
6.33	Descarte de pacotes por falta de rota considerando arquivos de 100 KB. . .	63
6.34	Descarte de pacotes por falta de rota considerando arquivos de 800 KB. . .	64

6.35	Sobrecarga considerando arquivos de 100 KB. . . . .	64
6.36	Sobrecarga considerando arquivos de 800 KB. . . . .	65
6.37	Atraso considerando arquivos de 100 KB. . . . .	65
6.38	Atraso considerando arquivos de 800 KB. . . . .	66
6.39	Taxa de Entrega considerando arquivos de 100 KB. . . . .	66
6.40	Taxa de Entrega considerando arquivos de 800 KB. . . . .	67
6.41	Vazão Efetiva considerando arquivos de 100 KB. . . . .	67
6.42	Vazão Efetiva considerando arquivos de 800 KB. . . . .	68
6.43	Descarte de pacotes considerando arquivos de 100 KB. . . . .	68
6.44	Descarte de pacotes considerando arquivos de 800 KB. . . . .	69
6.45	Descarte de pacotes por falta de rota considerando arquivos de 100 KB. . .	69
6.46	Descarte de pacotes por falta de rota considerando arquivos de 800 KB. . .	70
6.47	Sobrecarga considerando arquivos de 100 KB. . . . .	70
6.48	Sobrecarga considerando arquivos de 800 KB. . . . .	71
7.1	Avaliação da DHT. . . . .	74

## LISTA DE TABELAS

4.1	Tabela de rotas virtuais do VTMP. . . . .	24
5.1	Quadro geral dos cenários de simulação. . . . .	30
5.2	Cenários de simulação do tempo de atualização de rota. . . . .	31
6.1	Cenários de simulação do TCP. . . . .	42

## LISTA DE ABREVIATURAS E SIGLAS

<b>ACK</b>	<i>Acknowledgments</i>
<b>AODV</b>	<i>Ad Hoc On-Demand Distance Vector</i>
<b>C<sup>3</sup>TCP</b>	<i>Cross-layer Congestion Control for TCP</i>
<b>CCC</b>	Cubo de Ciclo Conexo
<b>DHT</b>	<i>Distributed Hash Table</i>
<b>DSDV</b>	<i>Destination Sequenced Distance Vector</i>
<b>DSR</b>	<i>Dynamic Source Routing</i>
<b>FTP</b>	<i>File Transfer Protocol</i>
<b>IP</b>	<i>Internet Protocol</i>
<b>MAC</b>	<i>Media Access Control</i>
<b>MANET</b>	<i>Mobile Ad Hoc Network</i>
<b>NJ</b>	<i>New Jersey</i>
<b>NS-2</b>	<i>Network Simulator</i>
<b>OLSR</b>	<i>Optimized Link State Routing</i>
<b>PAR</b>	Pacote de Atualização de Rota
<b>PC</b>	Pacote de controle
<b>PD</b>	Pacote de Dado
<b>PDA</b>	<i>Personal Digital Assistants</i>
<b>PRR</b>	Pacote de Resposta de Rota
<b>RoR</b>	<i>Rings of Rings</i>
<b>RREP</b>	<i>Route Reply</i>
<b>RREQ</b>	<i>Route Request</i>
<b>RTT</b>	<i>Round Trip Time</i>
<b>SMSS</b>	<i>Sender Maximum Segment Size</i>
<b>SSTHRESH</b>	<i>Threshold Size</i>
<b>TCP</b>	<i>Transmission Control Protocol</i>
<b>TCP-GAP</b>	<i>TCP with Gateway Adaptive Pacing</i>
<b>TCP-WOW</b>	<i>TCP With One Way</i>

<b>TTL</b>	<i>Time to Live</i>
<b>VTMP</b>	<i>Virtual Transparent Mobility Protocol</i>
<b>ZRP</b>	<i>Zone Routing Protocol</i>

# CAPÍTULO 1

## INTRODUÇÃO

As redes Ad Hoc Móveis (*Mobile Ad Hoc Networks* (MANETs)) são um tipo específico de rede sem fio que não depende de uma infra-estrutura pré-existente para funcionar [3, 36]. Uma MANET é composta por dispositivos móveis heterogêneos que normalmente são alimentados por baterias, podendo ser formada por notebooks, *Personal Digital Assistants* (PDAs), celulares, entre outros. Essas redes não dependem de uma administração centralizada, são um sistemas distribuído e autônomo [3].

As MANETs utilizam o meio sem fio para a comunicação, sendo possível o emprego de qualquer padrão. Por exemplo o Wi-Fi [40], o Bluetooth [40] e o infravermelho [44]. Outra característica destas redes é que não existe uma infra-estrutura fixa para prover a interação entre os nós, a comunicação ocorre diretamente ou depende da colaboração de nós intermediários [23]. Deste modo, os protocolos de roteamento em MANETs são distribuídos, colaborativos e multi-saltos.

As MANETs foram idealizadas e projetadas para fins específicos e temporários, apresentando muitos benefícios e facilidades: são auto-organizáveis, possuem baixo custo e de fácil implantação, permitem uma maior mobilidade e flexibilidade para os usuários [3]. Para fins militares as MANETs podem ser utilizadas nas situações de combate em território inimigo [36, 41]. As MANETs também podem ser usadas em situações de emergência [33], quando a infra-estrutura de comunicação foi destruída. Outra área em que estas redes são estudadas e aplicadas é no sensoriamento [36, 30], coletando informações em lugares onde não há uma estrutura de comunicação para estes fins.

Dadas as características e possibilidades que as MANETs apresentam, essas redes passaram a ser um grande foco de pesquisas. Com o aumento da capacidade, tanto de processamento como energética dos dispositivos móveis, as pesquisas começaram a se expandir para novas áreas [12], tais como: a criação de redes espontâneas, aplicação na telefonia celular e como uma extensão da Internet [3, 36, 12]. Por meio da colaboração dos dispositivos, pode-se garantir uma maior cobertura para a Internet, mobilidade e conectividade para os usuários, sem a necessidade

de investimentos adicionais de infra-estrutura fixa. As MANETs se apresentam como uma ferramenta que atende à necessidade crescente das pessoas de estarem cada vez mais conectadas à Internet, em qualquer lugar, buscando a mobilidade aliada à conectividade.

## 1.1 Problema e Objetivo

Para que o uso das MANETs se torne viável, ele ainda depende de padronizações e protocolos capazes de garantir segurança, conectividade, interoperabilidade, entre outras questões [3]. As MANETs apresentam ainda limitações de escalabilidade, qualidade de serviço, segurança e energia, além da mobilidade dos dispositivos. A mobilidade torna a topologia totalmente dinâmica, há a quebra de enlaces e pode ocorrer o isolamento de partes da rede. Esses fatores afetam o projeto da pilha de protocolos do modelo TCP/IP, para que seja possível uma rápida adaptação às mudanças de topologia.



Figura 1.1: Pilha de protocolos modelo TCP/IP.

O modelo TCP/IP define uma pilha de protocolos que permite a interconectividade entre diferentes equipamentos. Ele é composto por diferentes camadas, cada uma responsável por determinados serviços. Na Figura 1.1 é representado o modelo de camadas TCP/IP, pode observar-se que ele é composto por: uma camada de aplicação, onde reside as aplicações de rede e seus protocolos; uma camada de transporte, responsável pelo transporte de mensagens entre uma origem e um destino; uma camada de rede, responsável pela movimentação de pacotes pela rede, que determina por qual caminho o pacote irá seguir da origem até o destino; uma camada de enlace, responsável pelo acesso ao meio, auxiliando a camada de rede no envio de dados de

um nó para o próximo nó de uma rota; uma camada física responsável pela transmissão dos dados, no nível físico, entre dois nós [29].

As MANETs, assim como as demais redes, podem seguir o modelo TCP/IP para funcionar. Sendo assim, para a utilização de serviços da Internet em uma MANET, espera-se que os protocolos e aplicações que são executadas na Internet funcionem corretamente nesta rede, sem grandes alterações. Um dos problemas encontrados para que isso ocorra é a topologia dinâmica da rede, que afeta negativamente muitos protocolos e aplicações utilizados atualmente na Internet. Um exemplo deste problema é o *Transmission Control Protocol* (TCP), o qual sofre grande interferência com a mobilidade [3, 43, 25].

O protocolo TCP é um protocolo da camada de transporte, e foi projetado para criar uma conexão confiável sobre uma estrutura de rede não confiável, sendo planejado para ser independente da camada de rede, mas a realidade das implementações é outra [44]. A maioria das implementações do TCP foi desenvolvida e otimizada considerando que praticamente não há perda de pacotes na camada física, assim sendo, as perdas de pacotes são atribuídas ao congestionamento da rede [34]. O TCP utiliza confirmações de pacotes, *acknowledgments* (ACK), duplicadas e *Time Outs* de pacotes para identificar o congestionamento. Quando é detectado o congestionamento diminui-se a taxa de transmissão, evitando sobrecarregar a rede. Considerando-se uma rede cabeada, essa asserção é correta, as implementações interpretam corretamente o congestionamento, na grande maioria dos casos [44].

Entretanto quando se utiliza o meio sem fio para a comunicação de dados, há muito mais interferência e ruído do que nas redes cabeadas. Estas interferências ocasionam uma perda maior de pacotes no nível físico, sendo que essas perdas são interpretadas como congestionamento pelo TCP [34, 44]. Então o TCP reduz a transmissão de pacotes, diminuindo assim, a taxa com que os pacotes de dados são enviados pela rede. Nessas situações a ação deveria ser oposta, retransmitindo o quanto antes os pacotes perdidos para que o desempenho do TCP não fosse tão afetado [44].

Nas MANETs esse problema é agravado, pois além do uso das redes sem fio para a comunicação, as unidades são móveis e podem ocorrer perdas de pacotes por conta da quebra de rotas devido a mobilidade. As soluções propostas para esses problemas sugerem alterações no TCP [20, 31, 16], ou que as aplicações ignorem o TCP e incorporem suas funções diretamente [4]. Por exemplo, em [48] foi proposto a aplicação MADPastry, que é a implementação do protocolo

de aplicação *Distributed hash table* (DHT). As DHTs convencionais são implementadas e executadas na camada de aplicação e são independentes das camadas inferiores, apenas utilizando os serviços dessas camadas para mandar e receber informações de outros nós pertencentes a DHT. O trabalho desenvolvido em [48] sugere que: Para a DHT funcionar sobre uma rede MANET, é necessário que a DHT conheça a mobilidade dos nós, caso contrário ela não funcionará adequadamente. A solução proposta pelos autores foi implementar a DHT na camada de roteamento, modificando o funcionamento desta camada. O MADPastry, alcançou resultados significativos, mas tendo que modificar o funcionamento da DHT.

Realizar modificações nas aplicações ou no protocolo de transporte não é tão simples, pois envolve não apenas uma aplicação ou um protocolo, mas sim várias aplicações, o que torna as modificações muito custosas. O ideal seria apenas executar nas MANETs as aplicações desenvolvidas para a Internet, sem alterações nelas ou no protocolo de transporte, reduzindo os custos e facilitando a utilização. Nesses casos, mecanismos adicionais são necessários para permitir o funcionamento dos protocolos tradicionais de transporte nas MANETs.

Para tanto neste trabalho é proposta a criação do Protocolo Virtual de Transparência da Mobilidade (*Virtual Transparent Mobility Protocol* (VTMP)), visando reduzir os problemas que ocorrem com o TCP nas MANETs. Por meio de uma topologia virtual fixa esconde-se a mobilidade e deste modo permitindo a utilização de aplicações desenvolvidas para a Internet diretamente nas MANETs, sem a necessidade de alterações nos protocolos de transporte ou nas aplicações.

Outra característica importante do VTMP é que sua implantação não requer alterações na camada de rede ou transporte; pois ele atua como protocolo de manutenção da rede fixa entre essas camadas, encaminhando as mensagens por meio dessa rede virtual. O VTMP não está atrelado ao protocolo de roteamento, com isso pode ser utilizado sobre qualquer protocolo da camada de rede.

O VTMP foi simulado sob diferentes implementações do protocolo de transporte TCP, avaliando seu desempenho em diferentes cenários. Os resultados alcançados nas simulações demonstraram que é possível a criação de uma rede virtual fixa sobre a rede móvel. O uso do VTMP melhorou a taxa de entrega e reduziu os descartes de pacotes nos cenários em que a densidade e a velocidade de movimentação dos nós é baixa. A desvantagem do uso do VTMP está no aumento da sobrecarga de pacotes, gerada na manutenção das rotas da estrutura virtual

e no aumento do atraso, gerado pelas rotas mais longa em alguns casos.

## 1.2 Estrutura do Trabalho

O restante deste trabalho está dividido da seguinte maneira: No capítulo 2, é apresentada a fundamentação dos protocolos utilizados no trabalho, sendo descritos os protocolos de roteamento e de transporte. No capítulo 3 é apresentada uma revisão dos trabalhos que buscam melhorar o desempenho do TCP e das aplicações sobre as redes sem fio. No capítulo 4 é apresentado o VTMP, sendo detalhadas suas estruturas e o seu funcionamento. No capítulo 5 é apresentado o simulador utilizado, a definição dos parâmetros, bem como os cenários para as simulações. No Capítulo 6 são apresentados os resultados das simulações do VTMP, destacando os resultados alcançados sob diferentes cenários e implementações do TCP. No capítulo 7 é apresentado o resultado da execução de uma aplicação real sobre o VTMP. Por fim, no capítulo 8 são apresentadas as conclusões e os trabalhos futuros.

## CAPÍTULO 2

### FUNDAMENTOS

Neste capítulo são apresentados os protocolos da camada de rede e da camada de transporte, por serem as camadas que estão diretamente relacionadas com o protocolo VTMP. Nas próximas seções estão descritos os protocolos de roteamento, que fazem parte da camada de rede, e as diferentes implementações do protocolo TCP, que fazem parte da camada de transporte.

#### 2.1 Protocolos de Roteamento

Os protocolos de roteamento são responsáveis por estabelecer e manter as rotas para troca de informações entre os nós de uma rede. As características das MANETs de mobilidade e topologia dinâmica causam constantes desconexões, quebras de rotas, perdas de dados e sobrecarga em pontos da rede. Devido a estas características são necessários protocolos de roteamento robustos e com bom desempenho sobre as redes dinâmicas [3]. Os protocolos também devem ser simples, fáceis de implementar e não utilizar muitos recursos, devido às limitações de energia e de processamento dos dispositivos.

Para melhor atender aos requisitos e características das MANETs, diferentes protocolos de roteamento foram desenvolvidos [2, 9, 19, 23, 37, 21, 38, 35, 6, 39, 27, 5, 47], cada um com suas particularidades. Os protocolos de roteamento se diferem entre si, buscando o melhor equilíbrio entre os gastos com o gerenciamento da rede e os gastos com o tráfego de dados [3]. Sendo assim os protocolos podem ser agrupados conforme suas características de funcionamento. Uma das principais classificações utilizada é a divisão em protocolos pró-ativos, protocolos reativos e protocolos híbridos [36].

Os protocolos pró-ativos são aqueles que mantêm rotas atualizadas para os demais nós da rede e quando dois nós desejam se comunicar a rota já está estabelecida [36]. Como exemplos deste tipo de protocolo, pode-se citar o *Destination Sequenced Distance Vector* (DSDV) [37] e o *Optimized Link State Routing* (OLSR) [9]. Os protocolos pró-ativos apresentam como desvantagem um maior tráfego de controle, devido às constantes atualizações das rotas [36, 3].

Nos protocolos reativos, uma rota entre dois nós é criada quando há uma solicitação de

criação de rota, ou seja, quando um nó deseja enviar algo para outro nó [36]. Como exemplos desse tipo, pode-se citar o protocolo *Ad hoc On-Demand Distance Vector* (AODV) [38] e o protocolo *Dynamic Source Routing* (DSR) [23]. Os protocolos reativos apresentam uma latência maior para o envio de dados, pois a rota precisa ser criada antes do envio, mas apresentam como vantagem uma menor sobrecarga na rede, por não criar rotas que talvez não serão usadas.

Os protocolos híbridos são os que agregam características pró-ativas e reativas. Para uma vizinhança local (por exemplo a dois saltos de distância), cada nó mantém rotas atualizadas pró-ativamente, com mensagens de controle destinadas apenas à vizinhança local. Para nós que estão fora da vizinhança local as rotas são criadas de modo reativo, apenas quando elas forem solicitadas. Um exemplo de protocolo híbrido é o *Zone Routing Protocol* (ZRP) [19]. O desempenho dos protocolos híbridos é intermediário entre os pró-ativos e reativos.

O protocolo VTMP proposto neste trabalho, é independente da camada de rede, deste modo pode ser utilizado sobre qualquer protocolo de roteamento. Para as avaliações do VTMP foi escolhido o protocolo AODV, um dos protocolos mais utilizados nas MANETs.

### 2.1.1 *Ad Hoc On-Demand Distance Vector*

O protocolo de roteamento AODV [38] é um protocolo reativo, o qual cria rotas apenas quando recebe uma requisição de transmissão de dados. As rotas ficam ativas enquanto estiverem sendo utilizadas ou por um tempo pré-determinado.

As rotas são mantidas nas tabelas de roteamento, em que cada linha da tabela armazena informações para se chegar ao próximo nó da rota. Esta tabela é utilizada para encaminhar os pacotes que chegam ao nó, verificando quem é o destino e se na tabela existe informações de como se chegar ao destino. Caso exista um registro de como se chegar ao destino o pacote é enviado para o próximo nó; caso contrário o processo de descoberta de rota é iniciado.

O processo de descoberta de rota consiste no envio de um pacote de pedido de rota denominado *Route Request* (RREQ). O pacote RREQ é enviado em *broadcast* na rede até chegar ao destino. Cada RREQ gerado pela origem é identificado unicamente por uma dupla (endereço da origem, identificador de *broadcast*). Cada nó ao receber um RREQ segue os seguintes passos:

1. Verifica se o pacote não foi processado anteriormente, por meio da dupla (endereço da origem, identificador de *broadcast*). Se o pacote já foi recebido anteriormente ele é descar-

tado, evitando assim a formação de *loops* na rede.

2. Caso o pacote não tenha sido recebido, é verificado se o nó atual é o destino do RREQ. Caso o nó seja o destino, cria-se um pacote de resposta de rota, denominado *Route Reply* (RREP), e é enviado para a origem.
3. Caso contrário o RREQ é processado, adicionando-se um registro na tabela de roteamento. Define-se o endereço da origem do pacote como o endereço de destino, na tabela de roteamento, marcando como próximo nó o vizinho que enviou o pacote RREQ. Este processo cria a rota reversa, a rota do destino para a origem. Após ser registrado na tabela de roteamento o pacote é retransmitido em *broadcast*.
4. Quando o destino recebe o RREQ a rota reversa está formada, então o destino cria o pacote RREP, o qual cria a rota da origem para o destino. O RREP é enviado pela rota formada pelo RREQ. Cada nó intermediário ao receber o RREP registra em sua tabela de roteamento qual o próximo nó para se chegar ao destino. Assim, quando a origem receber o RREP, o caminho da origem para o destino estará formado e a comunicação estabelecida.

Em uma MANET pode ocorrer a quebra de um caminho. Neste caso o nó que identificar o erro da rota envia uma mensagem de erro para a origem do pacote indicando a ocorrência da quebra de rota. A origem então decide se irá ou não recriar esta rota.

A vizinhança local de um nó é conhecida quando este recebe uma transmissão de seus vizinhos. Esta conectividade local no AODV é utilizada para saber quais rotas de sua tabela ainda são válidas. Para manter atualizada a conectividade local, mensagens de HELLO também podem ser utilizadas. Quando um nó fica muito tempo sem transmitir pacotes ele envia uma mensagem de HELLO, avisando seus vizinhos que está ativo e que as rotas que passam por este nó ainda podem ser válidas. As mensagens de HELLO nunca são retransmitidas, sendo apenas utilizadas para o nó registrar sua vizinhança local.

## 2.2 Protocolo de Transporte

O protocolo de transporte TCP foi desenvolvido para a entrega de fluxos de dados fim-a-fim de modo confiável, sobre uma rede não confiável. O TCP é robusto para se ajustar às variações

de tráfego, diferenças de tamanhos de pacote e perdas de dados, entregando os dados de forma ordenada e sem erros para as aplicações.

O processo de funcionamento do TCP consiste em estabelecer uma conexão entre uma origem e um destino de um fluxo de dados da aplicação, definindo os parâmetros para esta comunicação. Após estabelecida a conexão, a origem divide o fluxo de dados em pacotes para serem enviados pela rede. O destino, ao receber os pacotes de dados, envia para a origem pacotes ACKs, informando qual foi o último pacote de dados sequencial recebido. Como a rede não é confiável, pode haver perda de pacotes, bem como os pacotes podem chegar fora de ordem. No primeiro caso, o TCP transmissor retransmite os pacotes faltantes, já no segundo caso, o TCP receptor ordena os pacotes recebidos fora de ordem e os entrega para a aplicação [44].

Baseadas na definição do TCP, descrito inicialmente na RFC 793 [24], diferentes implementações foram criadas e são utilizadas. Entre as implementações do protocolo TCP pode-se citar o TCP Tahoe [22], o TCP Reno [18], o TCP Vegas [7] e o TCP New Jersey [45]. Cada implementação visa melhorar o desempenho do TCP sobre as diferentes condições de tráfego de dados, deste modo, o estudo das características de cada implementação auxilia no entendimento do seu comportamento sobre o protocolo VTMP. A maior parte das implementações não foram desenvolvidas para MANETs, mas neste trabalho foram utilizadas implementações com e sem alterações para redes MANETs. A seguir é apresentada uma descrição mais detalhada de cada uma das implementações utilizadas neste trabalho, bem como são apresentadas as relações entre as implementações e as modificações propostas em cada uma das versões.

### 2.2.1 TCP Tahoe

Uma das primeiras implementações do protocolo TCP foi o TCP Tahoe [14], o qual é composto por três algoritmos: *Início Lento*, *Controle de Congestionamento* e *Retransmissão Rápida*. O TCP Tahoe foi o primeiro TCP a adicionar o algoritmo de *Retransmissão Rápida*, identificando e respondendo mais rapidamente à perdas de pacotes.

O processo de *Início Lento* é executado sempre no início da conexão ou quando ocorre o *Time Out*<sup>1</sup> de um pacote. Nesses casos, a janela de congestionamento<sup>2</sup> é definida com um tamanho máximo de segmento de envio (*Sender Maximum Segment Size* (SMSS)). O valor da

---

<sup>1</sup>Tempo máximo para a confirmação de entrega de um pacote é atingido.

<sup>2</sup>Número de pacotes que são transmitidos sem receber uma confirmação ACK dos pacotes enviados.

janela de congestionamento é aumentado exponencialmente até chegar a um tamanho limite, denominado *threshold size* (SSTHRESH), que determina o fim do *Início Lento* e o início do processo de *Evitar Congestionamento* [22].

O processo de *Evitar Congestionamento* consiste no aumento linear e não mais exponencial do tamanho da janela de congestionamento a cada recebimento de um ACK. Esse processo é executado até o recebimento de um ACK duplicado, ou até chegar ao limite estabelecido para a conexão.

O algoritmo de *Retransmissão Rápida* define que ao receber três ACKs duplicados, há uma grande chance do pacote ter sido perdido no meio de transmissão, então o pacote que recebeu os três ACKs é retransmitido. Neste caso o TCP reduz o SSTHRESH para a metade da janela de congestionamento, a janela de congestionamento é reduzida para 1 SMSS e o processo de *Início Lento* é executado. O algoritmo de *Retransmissão Rápida* não espera até ocorrer o *Time Out* do pacote para executar sua retransmissão, melhorando assim o tempo de resposta quando ocorre a perda de pacotes.

Na Figura 2.1 é apresentado o funcionamento da janela de congestionamento do TCP Tahoe. Nesta figura pode-se observar a influência de cada algoritmo no aumento da janela de congestionamento.

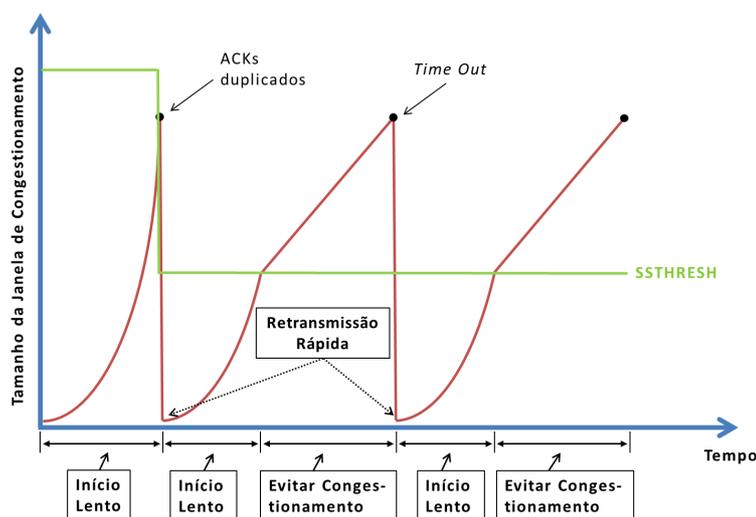


Figura 2.1: Janela de congestionamento TCP Tahoe [17].

## 2.2.2 TCP Reno

O TCP Reno e suas variações são as implementações do protocolo TCP mais utilizadas atualmente [18]. O TCP Reno possui 4 algoritmos, sendo 3 deles os mesmos do TCP Tahoe e adicionando o algoritmo *Recuperação Rápida*.

Os algoritmos de *Início Lento* e *Evitar Congestionamento* são os mesmos definidos na seção 2.2.1. Uma das grandes diferenças do TCP Reno, em relação ao TCP Tahoe, está na reação do algoritmo de *Retransmissão Rápida*. Nesta implementação foi definido que ao receber três ACKs duplicados, deve-se diminuir a janela de congestionamento pela metade e executar o algoritmo *Recuperação Rápida*. Em caso de um *Time Out* o processo de *Início Lento* deve ser executado.

O algoritmo de *Recuperação Rápida* consiste em reenviar o pacote faltante, o qual recebeu três ACKs duplicados, e esperar por um ACK de confirmação deste pacote. Caso não seja recebido um ACK em um tempo pré-determinado, executa-se o processo de *Início Lento*, reduzindo a janela de congestionamento para 1 SMSS, como ocorre no TCP Tahoe [14].

Na Figura 2.2 é apresentado o crescimento da janela de congestionamento do TCP Reno. A maior diferença em relação ao crescimento da janela de congestionamento do TCP Tahoe, está na execução do algoritmo de *Recuperação Rápida*.

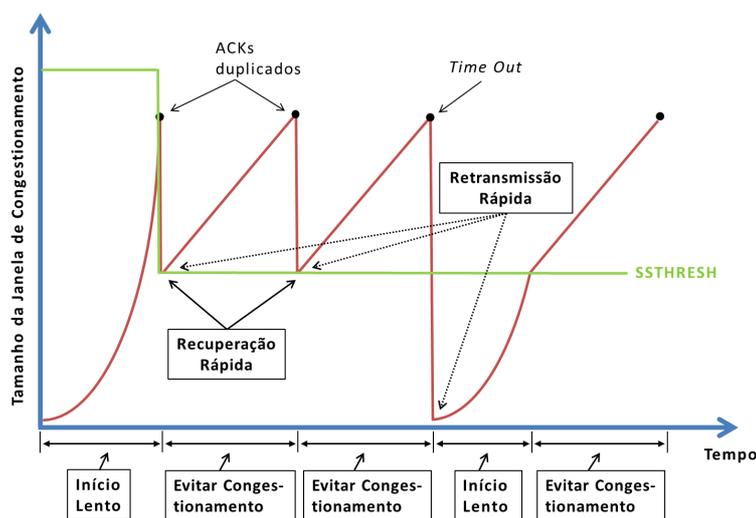


Figura 2.2: Janela de congestionamento TCP Reno [17].

O TCP Reno apresenta bom desempenho sob perdas de poucos pacotes, mas quando ocorrem perdas de pacotes consecutivos seu desempenho é afetado. Nesse caso o transmissor faz reduções sucessivas da janela de congestionamento até a confirmação de todos os pacotes perdidos. Com

isso, ocorre uma redução abrupta da taxa de transmissão. Outro problema do TCP Reno é que em caso de congestionamento, devido ao algoritmo *Recuperação Rápida*, ocorre uma perda maior de pacotes, já que o algoritmo demora mais para identificar o congestionamento [14].

### 2.2.3 TCP Vegas

O TCP Vegas é uma modificação da implementação do TCP Reno, melhorando sua reação a perdas sucessivas de pacotes e diminuindo as perdas por congestionamento. O TCP Reno é reativo ao congestionamento, enquanto o TCP Vegas é pró-ativo, tentando identificar o congestionamento para suavizar os seus efeitos [7].

As modificações realizadas no TCP Vegas abrangem alterações no processo de *Início Lento*. Essas alterações levam em conta que quando inicia-se o envio de dados, o algoritmo não conhece o estado da rede, não sabendo se está livre ou congestionada. O aumento exponencial da janela de congestionamento, pode levar ao envio de uma carga de dados superior a largura de banda disponível, ocasionando a perda de pacotes. Para contornar esse problema, o TCP Vegas aumenta exponencialmente o tempo de ida e volta do pacote *Round Trip Time* (RTT), até atingir um limiar pré-definido. Por meio do RTT é calculada a largura de banda disponível e definido o tamanho da janela de congestionamento.

O algoritmo de retransmissão do TCP Vegas é diferente das outras implementações do TCP:

- O TCP transmissor ao enviar um pacote mantém o tempo de envio (registro da hora em que o pacote foi enviado);
- O TCP transmissor ao receber um ACK, calcula, com base no tempo de envio registrado, quanto tempo levou para o pacote ser entregue e a confirmação retornar. O valor calculado é utilizado para manter o RTT estimado para os demais pacotes;
- Quando o TCP transmissor recebe um ACK duplicado, ele verifica se o tempo de envio do pacote é maior que o RTT estimado:
  - Caso isso ocorra: o pacote é retransmitido antes de receber três ACKs duplicados;
  - Caso contrario: é mantida a transmissão normal.
- Caso o TCP transmissor não receba um ACK, mas a diferença do tempo atual e o tempo de envio do pacote já passou o limiar RTT estimado, o pacote será reenviado sem ocorrer

um ACK duplicado.

Com esse algoritmo, tem-se uma recuperação muito mais rápida em caso de perdas de um único pacote ou de múltiplos pacotes, sem a espera por três ACKs duplicados [7].

O processo de *Evitar o Congestionamento* do TCP Vegas não usa a perda de pacotes para determinar se há congestionamentos ou não. No Vegas é calculada a taxa de envio e a taxa esperada; sempre que a taxa de envio for menor que a taxa esperada, aumenta-se a janela de congestionamento. Sempre que a taxa de envio for muito próxima da taxa esperada, diminui-se a janela de congestionamento, pois a largura de banda está sendo afetada pela quantidade de pacotes e com isso aumentando o tempo de entrega. Deste modo o Vegas consegue prevenir os efeitos do congestionamento antes que ele afete o envio dos dados, causando o descarte de pacotes.

Com essas modificações o TCP Vegas consegue melhores resultados em redes cabeadas do que o TCP Reno, contornando problemas de congestionamento e perdas de pacotes consecutivos que afetam o desempenho.

## 2.2.4 TCP New Jersey

Uma implementação do protocolo TCP que foi proposta para melhorar o desempenho nas redes sem fio é o TCP New Jersey [46]. Assim como o TCP Vegas, o New Jersey também usa uma abordagem pró-ativa ao congestionamento, mas foi pensado e direcionado para redes sem fio.

O TCP New Jersey é uma modificação sobre o TCP Jersey [45], melhorando a estimativa da largura de banda e do congestionamento na rede. Há poucas mudanças em relação aos processos de *Início Lento* e de *Evitar Congestionamento*, quando comparado ao TCP Reno. A grande diferença está no cálculo de estimativa da largura de banda e na identificação do congestionamento da rede.

No TCP New Jersey [46], o tamanho da janela de congestionamento é calculado dinamicamente e varia conforme a estimativa de largura de banda. O cálculo da largura de banda é realizado no receptor e enviado por meio de ACKs para o transmissor. O cálculo no receptor não é influenciado pela perda de ACKs e nem pelos atrasos na entrega destes pacotes. Com isso o resultado é mais preciso, baseado apenas no envio dos pacotes de dados.

Por meio desta estimativa de largura de banda o transmissor aumenta ou diminui sua janela

de congestionamento. Além da estimativa de largura de banda, o TCP New Jersey possibilita que os roteadores informem sobre congestionamentos na rede, marcando nos pacotes um bit que indica que a fila do roteador está acima de um limiar. O transmissor utiliza esta informação para diferenciar as perdas de pacotes devido ao congestionamento ou por erros, assim reagindo melhor a estas situações.

A reação do TCP New Jersey à perda de pacotes ocorre do seguinte modo: se um ACK for recebido com marcação de congestionamento, ele procede igual ao TCP Reno, iniciando o processo de *Início Lento* ou *Evitar o Congestionamento*, dependendo do tamanho da janela de congestionamento. Caso sejam recebidos três ACKs com a marcação de congestionamento, o tamanho da janela de congestionamento é recalculada e o algoritmo *Início Lento* ou *Evitar Congestionamento* é executado. Caso sejam recebidos três ACK duplicados sem a marcação de congestionamento, isso indica que o pacote foi descartado por um erro e o processo de *Recuperação Rápida* é iniciado.

## 2.3 Considerações Finais

Neste capítulo foram apresentados os protocolos que estão relacionados diretamente com a avaliação do protocolo VTMP. Na camada de rede qualquer protocolo de roteamento pode ser utilizado, neste trabalho foi definido o uso do protocolo AODV para a avaliação do VTMP, por ser um protocolo que gera pouca sobrecarga na rede e ser um dos protocolos mais utilizados nas pesquisas em MANETs. As implementações do protocolo TCP que serão utilizadas na avaliação do VTMP são a Tahoe Reno, Vegas e New Jersey. O uso de diferentes implementações visa avaliar o desempenho do VTMP sobre as diferentes versões já utilizadas do TCP, bem como sobre propostas de melhorias do mesmo.

## CAPÍTULO 3

### TRABALHOS RELACIONADOS

Neste capítulo é apresentada uma revisão dos trabalhos que propõem soluções aos problemas de funcionamento das implementações TCP sobre redes sem fio. Há diversos trabalhos descritos na literatura que alteram o protocolo de transporte TCP e sistemas da camada de aplicação.

#### 3.1 *Cross-layer Congestion Control for TCP* (C<sup>3</sup>TCP)

O protocolo *Cross-layer Congestion Control for TCP* (C<sup>3</sup>TCP) [26] foi desenvolvido para otimizar o uso dos recursos das MANETs, com o intuito de melhorar o desempenho do protocolo TCP. Segundo os autores, alguns dos problemas encontrados nas MANETs são a limitação da largura de banda, a restrição no tamanho das filas nos roteadores e o atraso na entrega dos dados. Para resolver esses problemas os autores propuseram uma abordagem *cross-layer*, fazendo a interação entre a camada de transporte e a camada *Media Access Control* (MAC) do protocolo de comunicação sem fio IEEE 802.11.

O C<sup>3</sup>TCP a camada MAC, ao receber um pacote TCP, calcula a largura de banda e o atraso do meio de comunicação, adiciona essas informações no pacote e o retransmite. Quando o pacote chega no destino, ele terá coletado informações sobre a largura de banda disponível e o atraso total do caminho percorrido do nó de origem até o nó de destino. O nó de destino então responde o pacote TCP com um pacote ACK que é enviado para o nó de origem.

O pacote ACK é encapsulado pela camada MAC, adicionando as informações de largura de banda do caminho e o atraso do pacote. Durante a propagação da ACK o atraso do caminho é somado. O nó de origem, ao receber o pacote ACK, segue uma abordagem *cross-layer* recuperando a informação que a camada MAC coletou no percurso. As informações coletadas (largura de banda do caminho e atraso) são utilizadas para calcular o tamanho mais adequado da janela de congestionamento do TCP. Deste modo a origem ao receber as informações de largura de banda e atraso consegue configurar as transmissões do TCP com os parâmetros calculados, evitando parte dos efeitos causados pelas redes sem fio multi-salto.

Esta solução, melhora a estimativa de congestionamento da rede e o desempenho do TCP,

mas ao custo de alterações em duas importantes camadas, o que demandaria mudanças significativas para que pudesse ser implantada.

### 3.1.1 *TCP with Gateway Adaptive Pacing* (TCP-GAP)

O protocolo *TCP with Gateway Adaptive Pacing* (TCP-GAP) [13] é uma solução criada para conectar uma rede sem fio multi-salto à Internet. O TCP-GAP apresenta um sistema de controle de congestionamento mais eficaz para as redes sem fio, melhorando o desempenho do TCP sobre as redes multi-salto.

A proposta dos autores altera a implementação do TCP que será executada na rede sem fio, sem alterar as implementações do TCP utilizados atualmente, mas mantendo a compatibilidade. Além da alteração no TCP também são necessárias alterações no nó *gateway* da rede, nó de ligação da rede sem fio com a Internet. A alteração do TCP é realizada no transmissor que está na rede sem fio, e essa alteração é dependente das modificações no *gateway*. O transmissor necessita saber a quantidade de saltos que o fluxo TCP percorre na rede sem fio e o tempo gasto para o pacote percorrer esse trajeto. Com base nessas informações e na largura de banda da rede o transmissor regula sua taxa de envio de dados, desse modo melhorando o desempenho do TCP.

Segundo os autores o *gateway* da rede deve monitorar e manter registros dos fluxos de dados TCP que passam por ele. Quando um pacote ACK entra na rede sem fio, as informações sobre o atraso e tamanho do caminho na rede sem fio são adicionadas no pacote, que posteriormente é recuperada pelo transmissor.

Uma outra função atribuída ao *gateway* é: Os fluxos de dados TCP que passam por um *gateway* podem gerar interferência uns nos outros, e um fluxo pode ocupar uma largura de banda maior que outro. Para melhor distribuir a largura de banda disponível entre os fluxos TCPs que passam pelo *gateway*, foi proposto um controle de fluxos, em que o *gateway* reduz a propagação de pacotes de um fluxo de dados maior, permitindo um aumento dos fluxos menores. Deste modo há uma melhor distribuição da largura de banda entre os nós pertencentes a rede. O sistema pode ser executado sobre qualquer protocolo de roteamento, desde que o mesmo permita a recuperação de informações sobre número de saltos na rede sem fio e a largura de banda disponível.

Com base nessas informações o TCP-GAP melhora o desempenho do TCP nas redes sem

fi, mas com o custo de mudanças na implementação do TCP do transmissor, que está na rede sem fio, e dos *gateways* que fazem a ligação da MANET com a Internet.

### 3.1.2 TCP Vegas MANET (TCP Vegas\_M)

O TCP Vegas MANET (TCP Vegas\_M) [15] é uma proposta de alteração do TCP Vegas para melhorar seu desempenho sobre uma MANET. Os autores propuseram um algoritmo que usa lógica difusa para estimar e diferenciar os estado da rede.

Na implementação do TCP Vegas\_M foram realizadas alterações apenas no algoritmo de controle de Congestionamento do TCP Vegas (descrição do TCP Vegas pode ser encontrada na seção 2.2.3), não alterando os demais algoritmos. A lógica difusa é utilizada para definir o estado atual da rede, por meio de um conjunto de treinamento, valores não lineares de entrada são mapeados para uma saída. Neste caso a entrada da lógica difusa é o número de nós percorridos pelo pacote e a variação do tempo RTT<sup>1</sup>. Com base nesses valores classifica-se o tráfego da rede como normal, congestionado, incerto ou com erros de bits de pacote. Com base nas saídas da lógica difusa, o controle de congestionamento é ajustado, diminuindo ou aumentando sua janela de congestionamento.

Para que o TCP tenha acesso ao número de nós por onde o pacote passou, é necessária uma interação entre a camada de transporte e a camada de rede, por meio dessa interação o TCP consulta o *Time to Live* (TTL) do pacote desta camada.

Os resultados apresentados pelos autores mostram uma melhora significativa no desempenho em comparação com a implementação padrão do TCP Vegas, mas ao custo de modificações nas implementações existentes do TCP.

## 3.2 TCP *With One Way* (TCP-WOW)

O TCP *With One Way* (TCP-WOW) [8] é uma implementação do TCP que altera o algoritmo de controle de congestionamento, visto que o TCP não consegue identificar corretamente o congestionamento em uma rede sem fio.

A proposta se concentra no cálculo de espaço disponível nas filas de pacotes da camada de rede pelos nós por onde o fluxo de dados TCP está passando. Para se chegar a este valor é

---

<sup>1</sup>Tempo para o pacote chegar ao destino e retornar a confirmação da entrega.

realizada uma estimativa do atraso para entrega de cada pacote no receptor do fluxo. O receptor então adiciona nos pacotes ACK o atraso calculado e os envia para o nó de origem. O nó origem utiliza esse valor calculado para definir o tamanho da janela de congestionamento e, deste modo, evitar uma redução abrupta do fluxo de dados enviado da origem para o destino em caso de perda de um pacote, sem que a rede esteja congestionada.

### 3.3 ADHOCTCP

O ADHOCTCP [34] é uma proposta de implementação do TCP desenvolvida especificamente para redes móveis, a qual modifica os algoritmos do TCP para melhorar o desempenho sobre essas redes. Os autores propuseram modificações no protocolo TCP, para que ele seja capaz de identificar o motivo da perda de pacotes, classificando em perda por congestionamento, por erro de canal ou por desconexão.

Com a informação do motivo da desconexão é possível implementar diferentes ações para cada caso. As ações descritas pelos autores são: iniciar o algoritmo para evitar o congestionamento, se a causa da perda de pacotes foi o congestionamento; retransmissão do pacote caso ocorra erro na rota; uso de mecanismos que permitam uma recuperação rápida quando ocorrer a reconexão da rota, caso venha a ocorrer uma desconexão.

Para o caso da identificação de congestionamento o ADHOCTCP depende do nó enviar uma notificação de congestionamento e com isso ele adota o controle convencional de congestionamento do TCP. Para saber se há uma quebra de rota, segundo os autores, depende-se de uma notificação explícita do protocolo de roteamento DSR para saber quando ocorreu uma quebra de rota.

A proposta do ADHOCTCP além de fazer alterações do TCP, não sendo mais compatível com as versões atuais, ele altera a camada inferior tirando a independência das camadas. As características do ADHOCTCP dificultam a adoção do protocolo.

### 3.4 Considerações Finais

Neste capítulo foi apresentada uma revisão da literatura, dos trabalhos que propõe o uso das aplicações da Internet sobre as MANETS. Os trabalhos encontrados na literatura em sua maioria propõe alteração nos protocolos existentes, dizendo não ser possível o uso dos protocolos

existentes sem alterações. A maioria das propostas são de alterações no protocolo TCP ou na camada de aplicação.

Neste trabalho é proposta uma abordagem diferente para resolução deste problema, a criação de uma camada virtual, que de modo transparente emule uma rede fixa sobre a rede móvel, permitindo o funcionamento dos protocolos de transporte tradicionais. No próximo capítulo é apresentado o protocolo proposto.

## CAPÍTULO 4

### VTMP – PROTOCOLO VIRTUAL DE TRANSPARÊNCIA DA MOBILIDADE

Este trabalho propõe um protocolo para tornar a mobilidade das MANETs transparente ao usuário, permitindo o uso de aplicativos de redes tradicionais sem alterações. O protocolo proposto é denominado Protocolo Virtual de Transparência da Mobilidade (*Virtual Transparent Mobility Protocol* (VTMP)). O VTMP foi projetado para atuar entre a camada de rede e a camada de transporte, sem fazer alterações nos protocolos destas camadas.

Na Figura 4.1 pode-se observar o posicionamento do VTMP na pilha de protocolos. A independência do VTMP em relação aos protocolos das camadas de rede e transporte proporciona uma fácil adequação, possibilitando sua execução sobre qualquer protocolo de roteamento.



Figura 4.1: Posicionamento do Protocolo VTMP.

O VTMP é composto por uma estrutura virtual (grafo) visível para o TCP como uma rede fixa, escondendo a mobilidade do protocolo de transporte e da camada de aplicação. Ao mesmo tempo, o VTMP deve garantir a conectividade desta rede fixa através da rede móvel e do protocolo de roteamento. Para tanto, periodicamente cada nó faz atualizações de suas rotas.

O VTMP é composto por 3 partes:

- A estrutura virtual, que define as ligações entre os nós;
- O módulo de atualização de rotas que mantém pró-ativamente as ligações virtuais da estrutura;
- Um módulo de roteamento que processa, encapsula e encaminha os pacotes de dados enviados pela camada de transporte.

No restante do texto adota-se o conceito de vizinho real e vizinho virtual de um determinado nó. Um vizinho real é um vizinho físico do nó que está no raio de transmissão do dispositivo. Neste caso os dois nós conseguem se comunicar diretamente sem a necessidade de qualquer nó intermediário entre eles. Dois nós são vizinhos virtuais quando existe uma ligação direta entre eles na estrutura virtual. Entretanto deve-se ressaltar que o enlace entre dois vizinhos virtuais pode ser constituído por diversos enlaces reais.

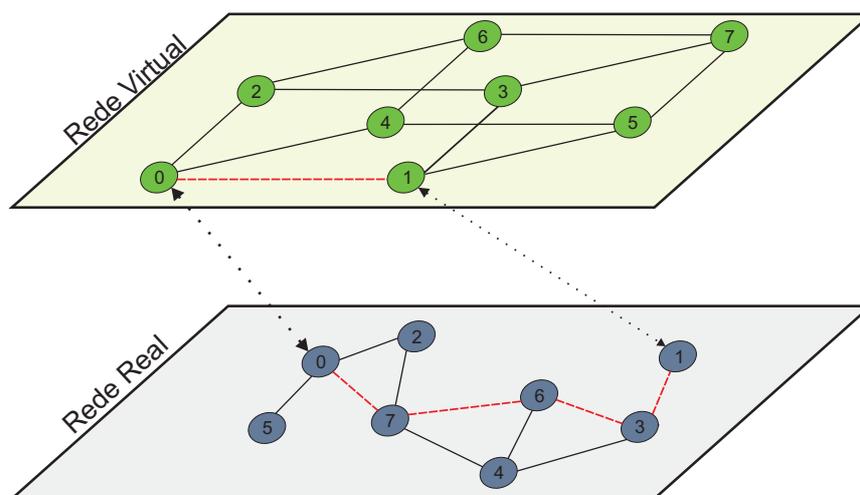


Figura 4.2: Rede virtual vs Rede real.

Na Figura 4.2 é representada a rede real e a rede virtual, para melhor exemplificar o conceito de vizinho real e virtual. Pode-se observar que na rede real o nó 1 tem como vizinho real o nó 3, mas na rede virtual seus vizinhos são os nós 0, 3 e 5, tendo enlaces diferentes do que na rede real. Na figura está destacado também o enlace do nó 1 com o nó 0, que na rede virtual estão a um salto de distância, porém quando traduzido para a rede real é um caminho de 4 enlaces reais entre eles.

## 4.1 Estrutura Virtual

A estrutura virtual do protocolo define a topologia da rede que será vista pelas camadas superiores, bem como as ligações virtuais entre os nós. Estas ligações determinam para quais vizinhos virtuais o nó deve manter rotas reais atualizadas, de modo pró-ativo. Todos os nós da rede devem conhecer a mesma estrutura virtual, para manter a coerência da topologia. A estrutura é fixa e dificilmente será alterada após a criação da rede. Para o correto funcionamento do VTMP, cada nó deve ter pré-carregado a estrutura virtual antes da formação da rede ou antes de ingressar na mesma.

A rede virtual é independente da rede física, sendo representada e armazenada em cada nó por um grafo bidirecional  $G = (V, E)$ . Onde  $V$  representa os nós que o grafo possui, e  $E$  representa os enlaces virtuais da rede, os quais são mantidos conexos pelo protocolo VTMP. Nessas condições, dado um enlace virtual  $(i, j) \in E$ , esse enlace é atualizado periodicamente por meio de um pacote de atualização de rota, mantendo o caminho ativo entre os dois nós.

A estrutura virtual da rede pode ser qualquer uma das estruturas: Anel de Anéis (*Ring of Rings* (RoR)) [2], Hipercubo [11], *Torus* [28] ou Cubo de Ciclos Conexos (CCC) [1]. Neste trabalho, após testes preliminares, a estrutura que demonstrou melhor resultado foi o Hipercubo, a qual foi utilizada nas demais avaliações.

Um Hipercubo de dimensão  $k$  é um grafo regular cujo conjunto de vértices é composto por  $N = 2^k$  vértices. De modo geral a construção de um Hipercubo de dimensão  $k$  ocorre do seguinte modo [11, 42]: Os  $2^k$  vértices são rotulados com seus respectivos números binários de 0 a  $2^k - 1$ ; existe uma aresta entre dois vértices  $s$  e  $p1$ , caso o rótulo de  $s$  divergir em apenas um bit do rótulo de  $p1$  (Ex:  $s = 000$  e  $p1 = 010$ ). A utilização do Hipercubo têm a vantagem da distância máxima entre dois vértices ser igual a  $\log_2 N$ , além de garantir a redundância de caminhos entre dois nós.

Na Figura 4.3 pode-se observar um exemplo de um Hipercubo com  $k = 4$  e  $N = 16$  nós. Este Hipercubo forma a topologia da rede virtual. Na figura 4.3 está ressaltado o nó  $s$ , em azul, que mantém ligações para os nós  $p1$ ,  $p2$ ,  $p3$  e  $p4$ , em verde. O nó  $s$  mantém pró-ativamente um caminho real para estes nós. Nesta topologia de rede, com 16 nós, cada nó mantém 4 rotas atualizadas. A distância máxima entre dois nós desta rede é 4.

Devido a redundância da estrutura virtual, o protocolo pode suportar a entrada e saída

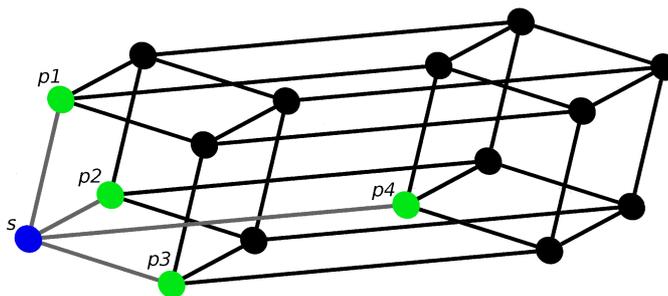


Figura 4.3: Exemplo de Hipergrafo.

limitada de nós da rede, desde que a estrutura virtual se mantenha conexa. Para suportar a entrada de novos nós, a estrutura virtual pode manter nós falsos, existentes apenas virtualmente. Com isso a rede continua funcionando normalmente, mesmo com a entrada de novos nós, os quais passam a operar com o valor de um desses nós falsos. O Protocolo também suporta a saída de nós, desde que, mesmo com a falta desses nós, a rede permaneça conexa.

Quanto maior for a redundância de caminhos virtuais menor é o efeito ocasionado pela entrada e saída de nós. Dadas as características das MANETs e do VTMP, quanto mais ligações um nó possui, maior será o número de atualizações de rota, podendo congestionar a rede apenas com pacotes de controle. Por isso, a escolha da estrutura da rede virtual é fundamental para garantir o funcionamento da rede sem prejudicar o tráfego de dados.

## 4.2 Descrição do VTMP

O VTMP cria uma rede virtual totalmente independente da rede física, utilizando uma estrutura de grafo, neste caso o Hipergrafo (Seção 4.1), a qual é armazenada em cada nó. Este grafo é utilizado para que os nós mantenham rotas atualizadas entre as ligações virtuais. A estrutura também é utilizada para enviar os pacotes de uma origem para um destino por meio da rede virtual, roteando as mensagens de dados.

Para a camada de transporte, o VTMP provê uma rede virtual fixa, encapsulando os pacotes do TCP e encaminhando-os pela estrutura virtual. Ao chegar no destino final do pacote o VTMP desencapsula o pacote e o entrega novamente à camada de transporte. Portanto, a rede conhecida pelo protocolo de transporte é a estrutura virtual fixa, e não a rede real.

Para o funcionamento do VTMP são utilizados dois tipos de pacotes: Os pacotes de controle e os pacotes de dados. Os pacotes de controle são para manter os enlaces entre os nós virtuais.

Os pacotes de dados são para o envio de pacotes do TCP.

Periodicamente cada nó deve verificar e atualizar as rotas para os seus vizinhos virtuais. Para a atualização de rotas, um pacote de controle é criado e enviado ao protocolo de roteamento, forçando a criação ou a atualização de uma rota para o nó desejado. Quando o nó recebe um pacote de dados, uma rota até o próximo nó virtual já está estabelecida, podendo ser utilizada sem maiores atrasos.

O VTMP mantém uma tabela de registros de rotas, diferente da tabela de rotas da camada de rede, na qual são mantidos temporariamente os enlaces que cada nó possui. Nesta tabela são mantidos os enlaces virtuais e rotas virtuais para os fluxos de dados: Os enlaces virtuais são registrados quando o VTMP recebe uma resposta de seu vizinho virtual, confirmando a criação de uma rota.

Quando um pacote de dados é enviado pela rota virtual o VTMP adiciona um registro temporário na tabela de roteamento que fica ativo enquanto o fluxo de dados estiver passando. Este registro auxilia o sistema a manter a coerência de caminho e não enviar os pacotes por diferentes rotas virtuais. Cada registro na tabela de rotas contém o endereço do nó de destino, o endereço do próximo nó virtual e o tempo de validade da rota. Na Tabela 4.1 é representada a tabela de rotas, com enlaces fictícios, apenas para exemplificar.

Tabela 4.1: Tabela de rotas virtuais do VTMP.

Destino	Próximo Nó	Tempo Validade
<i>a</i>	<i>a</i>	90.0
<i>d</i>	<i>g</i>	103.0
...	...	...

Os tempo de verificação e atualização dos registros de rotas são parâmetros de configuração do protocolo VTMP, podendo ser ajustados para o melhor funcionamento em relação ao protocolo de roteamento que esteja sendo executado sob o VTMP.

### 4.2.1 Pacotes

O VTMP envia dois tipos de pacotes diferentes, mas que possuem a mesma estrutura: os pacotes de controle e os pacotes de dados. Os pacotes de controle são divididos em dois tipos, os Pacotes de Atualização de Rota (PAR) e os Pacotes de Resposta de Rota (PRR), sendo que o campo de

dados desses pacotes é vazio. Os pacotes de dados contém, encapsulado no campo de dados, o pacote original enviado pelo TCP, o qual será desencapsulado e entregue no destino.

Origem	Destino	Tipo	Caminho
Dados			

Figura 4.4: Pacote do VTMP.

Na Figura 4.4 é apresentado o pacote do VTMP, o qual é composto pelos campos:

- **Origem** - Identificador do nó de origem do pacote.
- **Destino** - Identificador do nó de destino final do pacote.
- **Tipo** - Identifica que tipo de pacote está sendo transmitido, podendo ser pacote de dados, pacote PAR ou pacote PRR.
- **Caminho** - No campo caminho é armazenado uma lista com os nós virtuais por onde o pacote já passou, esta lista é utilizada na identificação de laços no caminho do pacote.
- **Dados** - O campo de dados é utilizado pelos pacotes de dados, para encapsular o pacote original e enviá-lo pela rede virtual.

### 4.2.2 Atualização e Manutenção das Rotas

A atualização de rotas acontece periodicamente, sendo que o intervalo entre essas atualizações pode ser configurado para melhor se ajustar as características do protocolo de roteamento. As atualizações de rotas de um nó são realizadas periodicamente, seguindo o tempo de validade da rota. Quando o tempo de validade da rota é menor que o tempo atual, um novo pacote PAR é gerado e enviado para o destino cuja rota expirou. Esse PAR força o protocolo de roteamento para criar ou atualizar uma rota real entre os dois nós.

O destino ao receber um pacote PAR, atualiza sua tabela de registro de rotas e envia um pacote PRR para a origem. A origem ao receber o PRR também atualiza sua tabela de registro de rotas, fechando o ciclo de atualização. Cada nó verifica periodicamente sua tabela de registro de rota, para remover os registros que tenham o tempo de validade expirado.

### 4.2.3 Roteamento de Pacotes

O VTMP encapsula os pacotes TCP recebidos e os envia pela rede virtual, até o destino final do pacote, onde ele é desencapsulado e entregue para o TCP receptor. Para o envio de pacotes de dados basta definir a rota virtual por onde o pacote será enviado. A tradução da rota virtual em uma rota real não necessita de novas trocas de mensagens para se chegar ao destino, pois o processo de manutenção mantém as ligações virtuais.

Quando o VTMP recebe um pacote de dados da camada de transporte, é verificado se existe uma rota virtual estabelecida entre a origem e o destino desse pacote. Caso exista, o pacote é encapsulado e enviado ao destino. Caso o nó não tenha uma rota virtual estabelecida com o destino, outro processo é iniciado:

1. Inicialmente o algoritmo calcula a menor rota virtual ativa até o destino. Caso exista mais de uma com a mesma distância ele escolhe aleatoriamente dentre elas;
2. Escolhida a rota virtual, ela é registrada na tabela de rotas e o pacote encaminhado para o próximo nó da rede virtual;
3. O próximo nó virtual, ao receber o pacote de dados verifica se é o destino final do pacote, caso não seja repete o passo 1;
4. O nó destino ao receber o pacote de dados do VTMP, desencapsula o pacote TCP e o entrega para a camada de transporte;
5. O Fluxo de pacotes ACK também é encapsulado e enviado pela mesma rota dos pacotes de dados enviados anteriormente.

### 4.2.4 Otimizações

Para melhorar o desempenho do protocolo VTMP, podem ser implementadas otimizações no funcionamento do protocolo, as quais são descritas a seguir.

Os pacotes VTMP de dados que passam por uma ligação virtual entre dois nós acabam passando por vários nós reais, como explicado anteriormente. Entre os nós reais por onde o pacote passou pode haver um que seja o destino final do pacote, ou conhecer o caminho até o destino final do pacote. Para otimizar a entrega de pacotes foram implementadas três verificações durante a passagem de um pacote VTMP pelos nós reais da rede:

- O nó real verifica quem é o destino final do pacote, com o intuito de ver se o pacote TCP que está sendo enviado pela rede virtual não é destinado a ele. Caso o pacote seja destinado a ele, a transmissão é interrompida, o pacote é processado e entregue à camada de transporte. Essa otimização evita que o pacote circule pela rede virtual até chegar a um nó que seja um vizinho virtual do destino, acrescentando custos adicionais.
- Outra verificação realizada é uma consulta a tabela de vizinhança do nó, para verificar se o nó atual não contém uma rota virtual para o destino final do pacote. Caso o nó tenha uma ligação virtual com o destino final do pacote e essa rota esteja ativa, o nó envia o pacote diretamente para o destino final do pacote. Com isso diminui-se o número de pulos virtuais e conseqüentemente o número de pulos reais.
- O Protocolo AODV envia mensagens de HELLO com alcance de um salto, essas mensagens são enviadas para descobrir quem são os vizinhos físicos do nó. O VTMP tira proveito dessas mensagens para complementar sua tabela de vizinhança, sem ter nenhum custo adicional. Mantendo uma lista de seus vizinhos, o nó ao receber um pacote que seja endereçado para um de seus vizinhos reais, antecipa a entrega do pacote, desviando-o e entregando-o ao vizinho destino.

Na topologia do Hipercubo, cada nó deve manter rotas entre as ligações definidas na estrutura do Hipercubo. Caso todos os nós enviassem atualizações entre suas ligações, o custo seria de  $N * k$ , deste modo no exemplo da Figura 4.3 cada nó teria que manter 4 rotas e seriam necessárias  $16 * 4 = 64$  mensagens de atualização. No exemplo pode-se observar que o nó  $s$  mantém rotas para os nós  $p1$ ,  $p2$ ,  $p3$  e  $p4$ , já os nós  $p1$ ,  $p2$ ,  $p3$  e  $p4$  também devem manter rotas para o nó  $s$ . Para diminuir a quantidade de pacotes de controle enviados, um nó só envia mensagens para metade de suas ligações virtuais e o algoritmo garante que todos os nós manterão as rotas necessárias para formar o Hipercubo. Quando uma mensagem PAR destinada ao nó chegar, ele atualiza sua tabela de rotas, deste modo reduz-se de 64 mensagens de atualização, para 32 mensagens.

## CAPÍTULO 5

### DEFINIÇÃO DAS SIMULAÇÕES

Neste capítulo é apresentada a definição da avaliação experimental do protocolo VTMP. O protocolo foi avaliado por meio de um simulador de redes, o NS-2, analisando o custo de se utilizar o VTMP sobre uma rede Ad Hoc Móvel. Os resultados são uma compilação da variação de tráfego, área de mobilidade e velocidade dos nós.

#### 5.1 Métricas Utilizadas

Para a avaliação de desempenho do sistema proposto, foram escolhidas as seguintes métricas: atraso, sobrecarga, taxa de entrega, vazão efetiva, descarte total de pacotes e descarte por falta de rota. Estas são definidas abaixo:

- O atraso é o cálculo do tempo gasto para um pacote de dados ser entregue, desde o momento em que o pacote é enviado até o momento que é recebido no destino, sendo representado em milissegundos (ms). Quanto menor for o atraso, menos tempo os pacotes levam para ir da origem até o destino.
- A taxa de entrega é a porcentagem de pacotes de dados recebidos corretamente pelo destino, desconsiderando pacotes repetidos. Quanto maior o número de pacotes entregues, menor é o gasto com retransmissões de pacotes, aumentando a eficiência do sistema.
- A sobrecarga consiste na quantidade total de pacotes de controle enviados para cada pacote de dado recebido corretamente. O valor da sobrecarga é expresso em (pacotes de controle/pacotes dados recebidos). Quanto menor a sobrecarga, menor é o custo dos nós para o envio de dados.
- A vazão efetiva consiste no total de dados enviados entre dois nós, quantidade de dados (apenas pacotes de dados, desconsiderando pacotes de controle) enviada de um nó para outro por segundo. Para o cálculo da vazão efetiva soma-se os bytes de dados que um nó enviou e divide-se pelo tempo de envio dos dados, tempo do início da transmissão até o

recebimento da última confirmação de entrega de dados, sendo os resultados representados em (Bytes/segundo (B/s)). Quanto maior for a vazão de dados, mais rapidamente a informação é transferida de um nó para o outro.

- Descarte total de pacotes se refere a todos os pacotes de dados descartados durante a simulação, independente do motivo do descarte. Os descartes podem ocorrer por diversos motivos, entre eles pode-se citar a falta de uma rota para o envio, erro na transmissão, fila de roteamento cheia, entre outros. Quanto maior o número de pacotes descartados maiores são os gastos com uma transmissão.
- Os descartes de pacotes por falta de rota também foram avaliados. O descarte por falta de rota é um descarte de pacote específico, que ocorre quando um nó não consegue estabelecer uma rota para a mensagem chegar ao destino.

## 5.2 Parâmetros de Simulação

O VTMP foi implementado e avaliado utilizando o simulador *Network Simulator* (NS-2), em sua versão 2.34 [32]. O NS-2 é um simulador de eventos discretos, desenvolvido para pesquisas em redes de computadores, possibilitando a criação de ambientes de redes cabeadas, redes sem fio e redes Ad Hoc. Possui as implementações de protocolos da camada física, roteamento, transporte e aplicação, dentre as quais a implementação dos protocolos de roteamento (AODV), as implementações do protocolo de transporte (TCP) e aplicações utilizados neste trabalho.

A avaliação experimental foi realizada considerando um ambiente móvel, com os nós apresentando diferentes velocidades. A variação da velocidades considerada foi de 2 m/s até 20 m/s. O padrão de movimentação que os nós seguem é o *Random Waypoint*.

Foram definidas duas áreas de movimentação dos nós, uma de 1500x300 metros e outra de 4500x900 metros. Foram usadas duas áreas diferentes para melhor avaliar o protocolo proposto em situações distintas: Uma rede densa com até 6 saltos entre dois nós (1500x300 metros); A outra rede esparsa com até 18 saltos entre dois nós (4500x900 metros). Em ambas as redes foi mantida a mesma quantidade de nós (128 nós).

A estrutura virtual utilizada no VTMP para a representação da rede fixa foi o Hipercubo. Como a rede real possui 128 nós cada um mantém conexão com 7 nós virtuais.

Os parâmetros gerais utilizados nas simulações são descritos na Tabela 5.1.

Tabela 5.1: Quadro geral dos cenários de simulação.

<b>Parâmetros</b>	<b>Valores</b>
Número de nós	128
Ligações virtuais	7
Área	4500x900 m e 1500x300 m
Raio de alcance da transmissão	250 m
Velocidade do nós	2, 4, 8, 12, 16 e 20 m/s
Tempo de pausa	10 s
Modelo de mobilidade	<i>Random waypoint</i>
Protocolo MAC	802.11
Protocolo de roteamento	AODV
Tamanho da fila de roteamento	50
Largura de Banda	11 Mb
Protocolo de Transporte	TCP
Janela de Congestionamento do TCP	Padrão
Gerador de tráfego	FTP
Tamanho dos arquivos	100 à 800 KB
Tamanho do pacote	1024 bytes
Rede Virtual	Hipercubo

Nas simulações variou-se a quantidade de conexões de dados estabelecidas para avaliar o impacto do aumento do número de usuário ativos na rede. Durante a simulação, uma conexão *File Transfer Protocol* (FTP) entre dois nós só ocorrerá uma única vez. Para apresentar os resultados considera-se que cada simulação mantém 10 e 20 conexões FTP simultâneas.

O tamanho de pacote escolhido foi de 1024 bytes. O tamanho dos arquivos a serem transmitidos por FTP foi definido de 100 KB a 800 KB. As conexões entre os nós são geradas aleatoriamente e são disparadas a cada 25 segundos, sendo realizados 4 ciclos de conexão, em cada ciclo são disparadas 10 ou 20 conexões, dependendo do cenário. Com esta variação buscou-se avaliar o impacto na rede do aumento de tráfego de dados e usuários ativos.

Todos os resultados obtidos e apresentados neste trabalho são a média de 35 simulações para cada uma das diferentes configurações criadas. Por se tratar de uma rede híbrida com o uso do VTMP, definiu-se um tempo de inicialização da rede, durante esse tempo são formados os caminhos virtuais. Durante o tempo de inicialização não são transmitidos dados, apenas pacotes de controle para a criação das rotas do VTMP, nas simulações foi considerado 10 segundos para a rede estar apta a enviar dados.

### 5.3 Definição do Tempo de Atualização de Rotas do VTMP

A estrutura virtual do VTMP define os caminhos virtuais da topologia da rede virtual, esses caminhos são criados na rede móvel por meio de pacotes de atualização de rotas, que são enviados periodicamente. Para tanto, foram avaliados diferentes tempos de atualização de rota, de modo a encontrar o que tenha menor impacto e não degrade o desempenho da rede. Na Tabela 5.2 estão definidos os parâmetros específicos utilizados nestas simulações, sendo que os parâmetros gerais foram definidos na Tabela 5.1.

Tabela 5.2: Cenários de simulação do tempo de atualização de rota.

Parâmetros	Valores
Área	4500x900 m e 1500x300 m
Protocolo de roteamento	AODV
Protocolo de Transporte	TCP Reno
Tamanho dos arquivos	100 à 800 KB
Número de conexões simultâneas	10
Intervalos de atualização de rota	45, 60, 90 e 180 segundos

Nestas simulações definiu-se o uso do protocolo TCP Reno, por ser uma das implementações do TCP mais utilizadas atualmente na Internet, bem como por ser uma implementação que não é otimizada para uso em redes sem fio.

Para avaliar o tempo de atualização de rota que apresenta melhor resultado foram utilizados diferentes cenários, variando-se o tamanho dos arquivos e a área de movimentação dos nós

Os gráficos com as duas áreas estão lado a lado para melhor visualização do impacto de diferentes densidades de nós na métrica avaliada. Os gráficos com os resultados da transmissão de arquivos de 100 KB e 800 KB, em algumas métricas, não estão na mesma escala, para possibilitar uma melhor visualização dos resultados. Cada figura contém a compilação de resultados de um cenário, sendo que cada linha representa um tempo de atualização de rota, e está representada em uma cor. As marcações verticais de cada ponto no gráfico são o desvio padrão da métrica avaliada, sendo os resultados a média de 35 simulações para cada ponto.

O Atraso é apresentado nas Figuras 5.1 e 5.2. Na Figura 5.1(a) pode-se observar que o menor atraso foi com o tempo de 45 segundos (s) entre as atualizações de rota, ficando em 116ms abaixo do tempo de atualização de 180s com os nós movendo-se a 20m/s. Com a velocidade de 4m/s o atraso dos tempo de 45s, 60s e 90s são iguais, já o atraso do tempo de atualização de 180s foi de 133ms a mais em relação aos demais tempos de atualização. Em média o tempo de 45s tem um

atraso menor neste cenário, mas ao considerar o desvio padrão pode-se dizer que não há uma diferença significativa entre os tempos de atualização de rota de 45s, 60s e 90s,

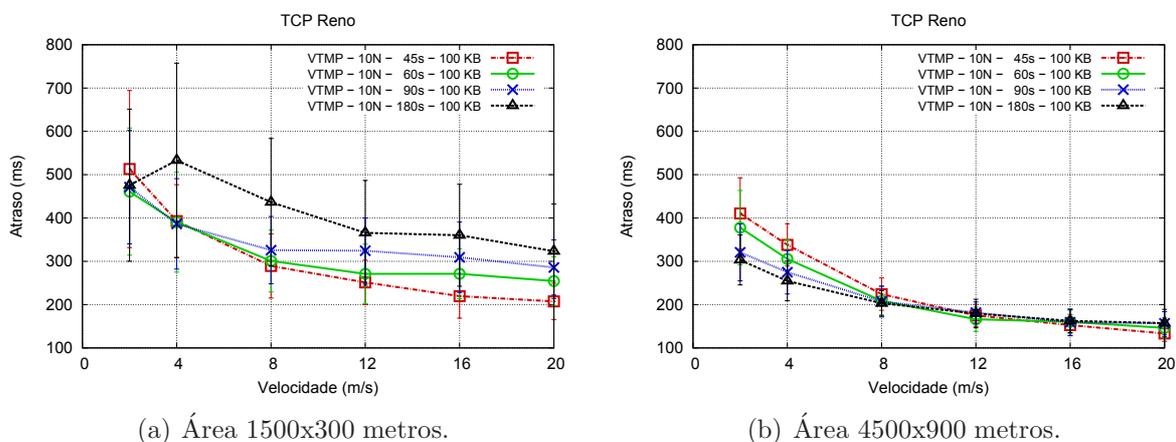


Figura 5.1: Atraso considerando arquivos de 100 KB.

Na Figura 5.1(b) observa-se que o atraso no cenário com a rede esparsa, quase não sofre influencia com os diferentes tempos de atualização de rota. Com baixa velocidade há uma pequena diferença nos atrasos, sendo que o tempo de atualização de 45s foi o maior, ficando 107ms acima do tempo de atualização que apresentou menor atraso (tempo de 180s), com os nós movendo-se a 2m/s. Pode-se observar ainda que não há diferença significativa na média dos atrasos quando a velocidade dos nós é igual ou superior a 8m/s.

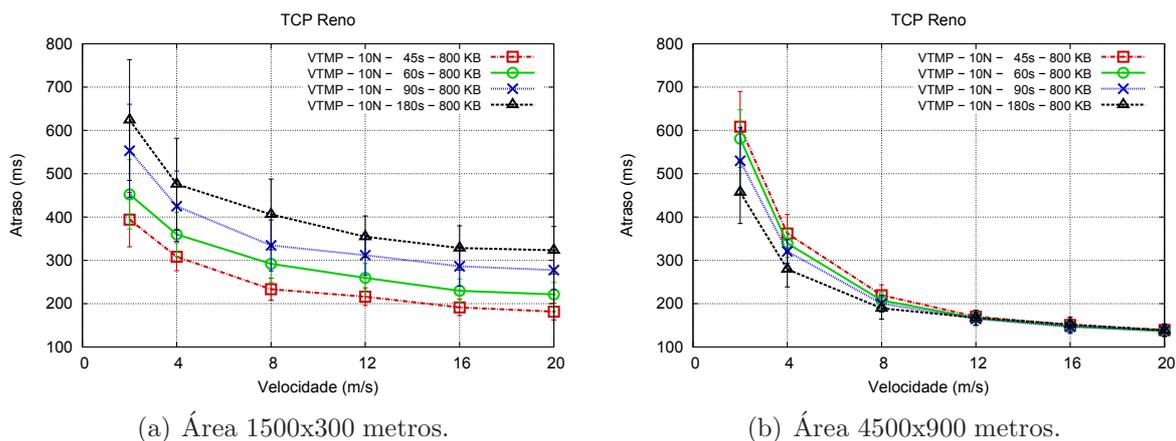


Figura 5.2: Atraso considerando arquivos de 800 KB.

Na figura 5.2(a) pode-se observar uma diferença mais significativa entre os diferentes tempos de atualização de rota. Neste cenário o tempo de atualização de rota que apresentou menor atraso foi o tempo de 45s, e o tempo de atualização que apresentou maior atraso foi 180s. A

diferença entre os tempos de atualização de 45 e 180s foi de 330ms, considerando a movimentação dos nós a 2m/s, e 142ms a mais considerando uma velocidade de 20m/s. Pode-se observar ainda que nesse cenário há um aumento linear no atraso da entrega dos pacotes, sendo que quanto maior o tempo entre as atualizações de rotas maior é o atraso.

No cenário apresentado na Figura 5.2(b), observa-se que não há uma diferença significativa no atraso dos tempos de atualização de rota, considerando-se o desvio padrão de cada ponto.

O tempo de atualização de rota que apresentou menor atraso na rede densa foi o tempo de 45 s, isso ocorre pois com atualizações frequentes garante-se que mais rotas estejam ativas, reduzindo o tempo de entrega dos pacotes. Pode-se observar que com o aumento da velocidade de movimentação há uma redução no atraso, isso ocorre pois com os nós movendo-se mais rapidamente há mais quebras de rota e a entrega de pacotes tende a ocorrer com um menor número de saltos. Deste modo, a entrega da maior parte dos pacotes ocorre com um tempo de atraso menor com velocidades mais altas.

Por meio dos resultados, observa-se ainda que há um comportamento inverso entre os tempos de atualização com relação a densidade de nós na rede. No cenário com uma área de 1500x300 m o tempo de atualização de 45 s apresentou menor atraso, já no cenário com área de 4500x900 m o tempo de atualização de 180 s apresentou menor atraso. Isso ocorre devido a rede de 4500x900 m ser esparsa e com isso ter poucos nós fazendo a ligação entre partes da rede, o que com o aumento da sobrecarga da rede usando o tempo de 45 s aumenta o tempo que os pacotes ficam na fila de roteamento.

No geral constatou-se que na maioria dos cenários não há uma diferença significativa entre os diferentes tempos de atualização de rota, na métrica de atraso. Apenas no cenário com a rede densa e arquivos de 800 KB, o tempo de 45s apresentou uma vantagem mais significativa, em comparação aos demais tempos de atualização de rota. Neste cenário o tempo de atualização de 180s apresentou o maior atraso, e os tempos de 60 e 90s não apresentaram uma diferença significativa entre si.

Nas Figuras 5.3 e 5.4 são apresentados os resultados de taxa de entrega de arquivos de 100 KB e 800 KB respectivamente. Na transmissão de arquivos de 100 KB, os tempos de atualização de rotas que tiveram melhor desempenho, considerando-se um cenário de alta densidade de nós (Figura 5.3(a)), foram os tempos de 90s e 180s, entregando 5% a mais de pacotes do que o tempo que obteve o pior resultado, no caso 45s. Neste cenário, apesar de na média o tempo de

180s ter uma taxa de entrega maior que o tempo de 90 s, não há diferença significativa entre os resultados ao considerarmos o desvio padrão dos resultados.

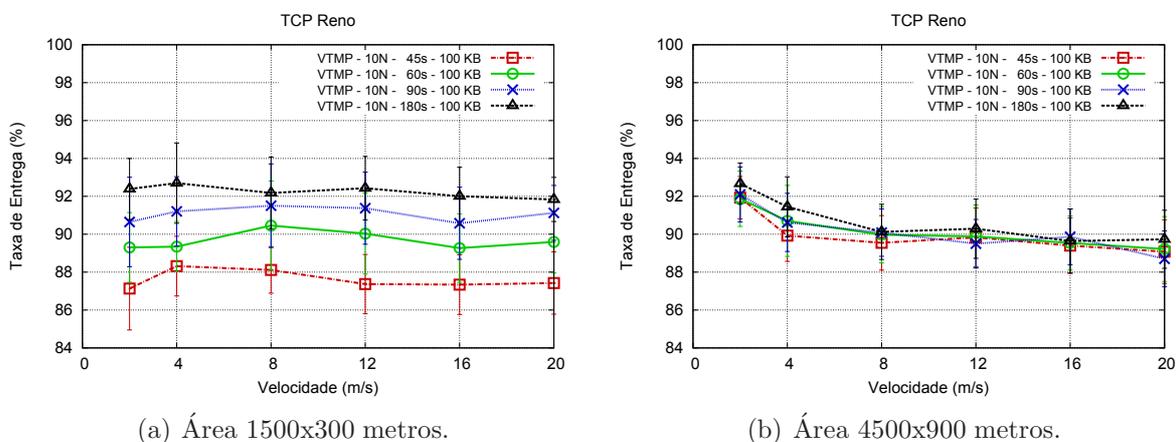


Figura 5.3: Taxa de Entrega considerando arquivos de 100 KB.

No cenário com a rede esparsa (Figura 5.3(b)), ao considerar o desvio padrão não há diferença significativa entre os tempos de atualização das rotas.

A taxa de entrega considerando-se arquivos de 800 KB em um cenário de alta densidade de nós é apresentada na Figura 5.4(a). Neste cenário os tempos de atualização de 90s e 180s foram os que apresentaram maior taxa de entrega, entregando 2% a mais de pacotes do que o tempo de 45 segundo, que apresentou a menor taxa de entrega. Novamente o tempo de 180s na média obteve maior taxa de entrega mas a diferença não é significativa ao considerar o desvio padrão dos tempos de atualização de 90s e 180s.

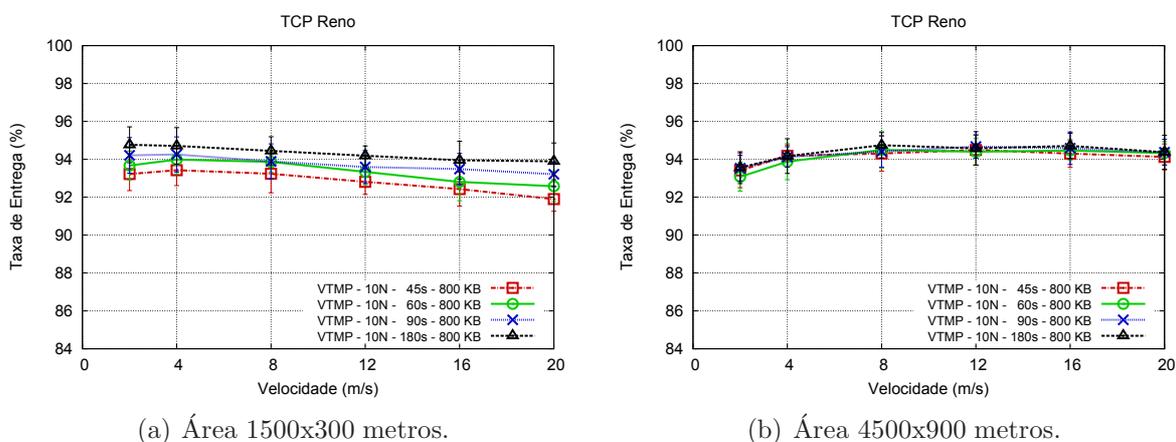


Figura 5.4: Taxa de Entrega considerando arquivos de 800 KB.

No cenário com a rede esparsa (Figura 5.4(b)), ao considerar o desvio padrão não há diferença

significativa entre os tempos de atualização das rotas.

Na taxa de entrega com redes esparsas a mudança do tempo de atualização não apresentou diferença significativa. Na rede densa a taxa de entrega diminuiu no tempo de atualização de 45s, isso ocorreu devido a alta sobrecarga nesse cenário, como pode ser observado nas Figuras 5.11 e 5.12, ocasionado pelas atualizações mais frequentes. Deste modo, na métrica de taxa de entrega, os tempos de atualização de rotas de 90 e 180s apresentaram resultados melhores, em cenários com alta densidade de nós, Figuras 5.3(a) e 5.4(a).

Os resultados da vazão efetiva são apresentados nas Figuras 5.5 e 5.6, com resultados da transferência de arquivos de 100 KB e 800 KB respectivamente.

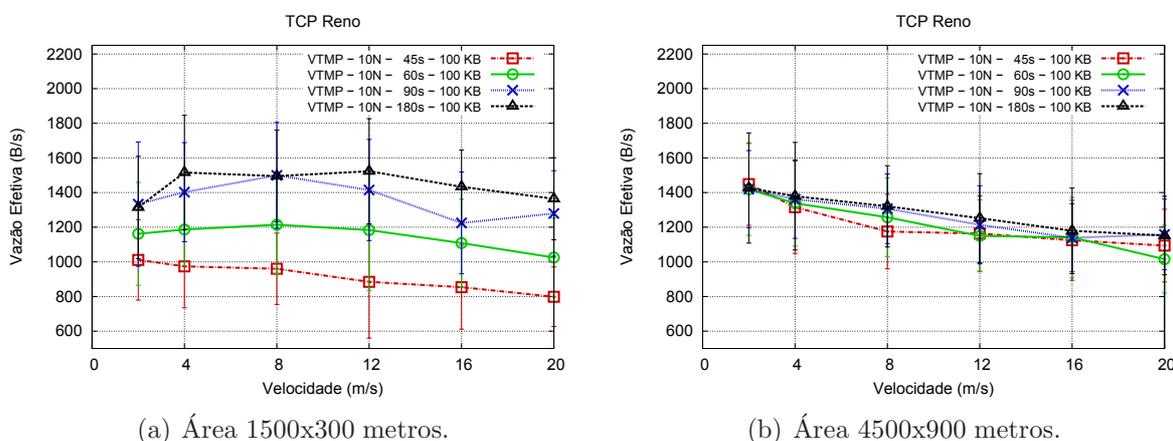


Figura 5.5: Vazão Efetiva considerando arquivos de 100 KB.

Pode-se observar, na Figura 5.5(a), que os tempos de atualização de rotas que obtiveram maior vazão foram os tempos de 90 e 180s. O tempo de atualização de rotas que apresentou menor vazão efetiva foi 45s, apresentado uma diferença de 303 B/s e 566 B/s, com os nós movendo-se a 2m/s e 20m/s respectivamente. Considerando-se o desvio padrão, o tempo de 90s e 180s, não apresentam uma diferença significativa nas diferentes velocidades de movimentação dos nós avaliadas.

A vazão no cenário com a rede esparsa apresentada na Figura 5.5(b), não apresentou diferença significativa, considerando-se o desvio padrão dos tempos de atualização de rotas.

Em um cenário mais denso, mas com transmissão de arquivos de 800 KB, os resultados são semelhantes aos obtidos nos cenários com transmissão de arquivos de 100 KB. Na Figura 5.6(a) pode-se observar que novamente os tempos de atualização de rota de 90 e 180s foram os que apresentaram maior vazão, com uma diferença em relação ao tempo de 45s, de 960 B/s e 1704

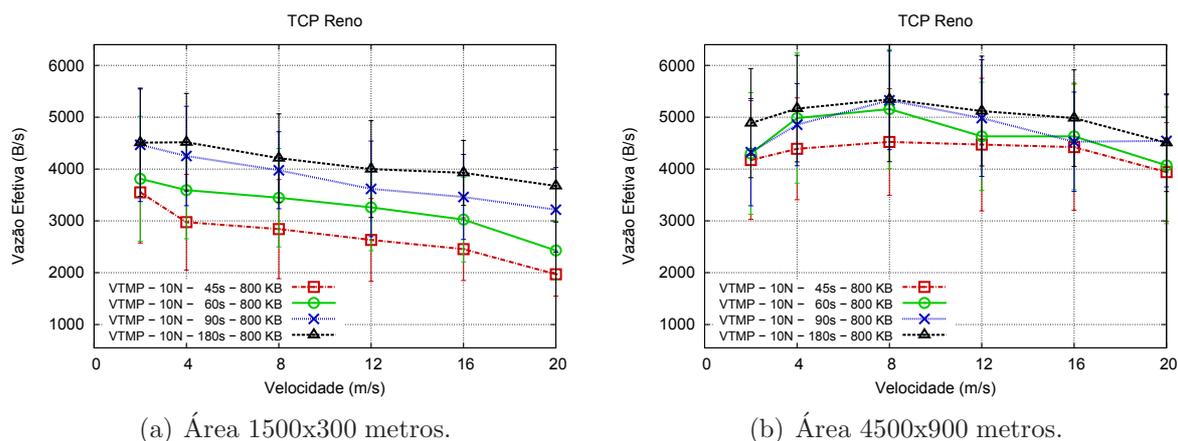


Figura 5.6: Vazão Efetiva considerando arquivos de 800 KB.

B/s com os nós movendo-se a 2m/s e 20m/s respectivamente. Pode-se observar também que os tempos de 90s e 180s não apresentam uma diferença significativa ao considerar o desvio padrão de ambos.

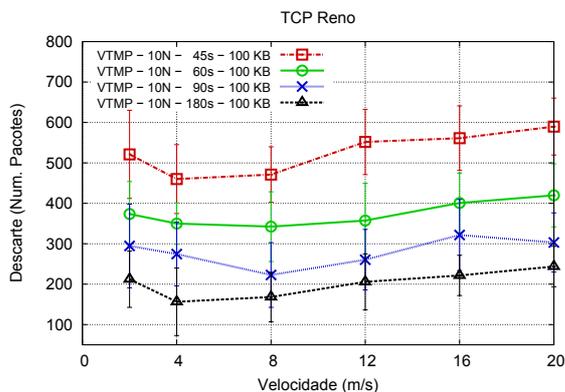
Como esperado, a vazão efetiva em arquivos grandes é maior do que com arquivos pequenos. A vazão efetiva foi maior na rede mais esparsa do que na rede densa, isso pode ser explicado pelo fato da rede densa apresentar uma sobrecarga maior e uma taxa de entrega menor, logo afetando negativamente a vazão efetiva.

No cenário com a rede esparsa, Figura 5.6(b), apesar de uma diferença na média entre os tempos de atualização, essa diferença não é estatisticamente significativa ao considerar o desvio padrão. Observa-se ainda, que há uma queda na vazão efetiva com o aumento da velocidade de movimentação de nós, isso ocorre pois quanto maior a velocidade mais quebras de rotas ocorrem, e o TCP mantém sua janela de congestionamento pequena, diminuindo assim a vazão.

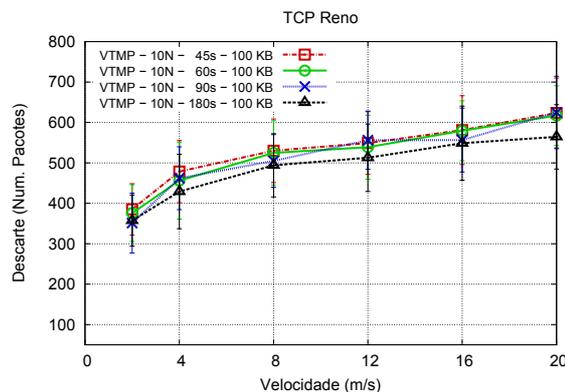
Na métrica vazão efetiva, os tempos de atualização que obtiveram melhor desempenho foram os tempo de 90 e 180s, ao considerar os cenários mais densos. Nos cenários com a rede esparsa não há diferença significativa entre os diferentes tempos de atualização de rota.

Nas Figuras 5.7 e 5.8 são apresentados os resultados de descarte total de pacotes de dados. Na área de 1500x300 metros com arquivos de 100 KB (Figura 5.7(a)) há diferença significativa entre os diferentes tempos de atualização de rota. O tempo 180s foi o que descartou menos pacotes e o tempo de 45s foi o que descartou mais pacotes. O tempo de 45s descartou em média 309 e 346 pacotes a mais que o tempo de 180s, com movimentação dos nós de 2m/s e 20m/s respectivamente. Os tempos de atualização de 90s e 60s foram intermediários entre os tempos

de 45s e 180s.



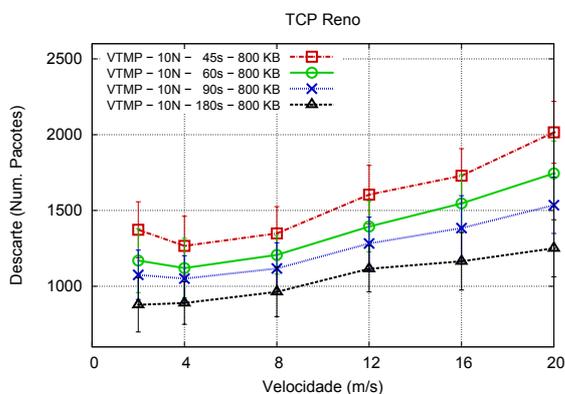
(a) Área 1500x300 metros.



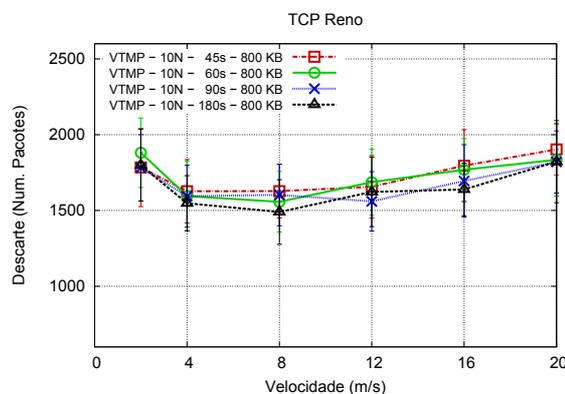
(b) Área 4500x900 metros.

Figura 5.7: Descarte de pacotes considerando arquivos de 100 KB.

O descarte de pacotes no cenário de 4500x900 metros (Figura 5.7(b)) não apresentou uma diferença significativa, sendo que o desvio padrão dos tempos de atualização não permitem uma conclusão mais precisa.



(a) Área 1500x300 metros.



(b) Área 4500x900 metros.

Figura 5.8: Descarte de pacotes considerando arquivos de 800 KB.

Considerando arquivos de 800 KB o resultado é similar, a transmissão de arquivos de 100 KB. O tempo de atualização que apresentou menor número de descarte de pacotes foi 180s, e o que apresentou maior número de descartes foi o tempo de 45s, conforme pode-se observar na Figura 5.8(a). Em média a diferença foi de 495 e 765 pacotes para uma velocidade de movimentação de 2m/s e 20m/s, respectivamente.

Nos resultados do cenário com a rede esparsa (Figura 5.8(b)) não há diferença significativa entre os tempos de atualização, pois os resultados estão muito próximos e o desvio padrão é

muito grande.

No cenário com alta densidade de nós, o tempo de atualização de rotas que apresentou melhor resultado foi o tempo de 180s, seguido por 90s, 60s e 45s. Nesta métrica pode-se observar que quanto menor o tempo entre as atualizações, maior o número de descartes de pacotes de dados.

O aumento nos descartes de pacotes pode ser explicado por meio da análise do gráfico de sobrecarga, em que o maior número de atualizações gera mais pacotes de controle na rede. Com mais pacotes transitando pela rede sem fio, as perdas de pacotes são mais frequentes, podendo ser ocasionadas por estouro de fila de roteamento, colisão ou falta de rota.

Os resultados de descarte de pacotes de dados por falta de rota no roteamento são apresentados nas Figuras 5.9 e 5.10. O descarte por falta de rota em uma área de alta densidade de nós não apresentou diferença significativa entre os diferentes tempos de atualização, tanto com a transmissão de arquivos de 100 KB (Figura 5.9(a)) quanto com a transmissão de arquivos de 800 KB (Figura 5.10(a)).

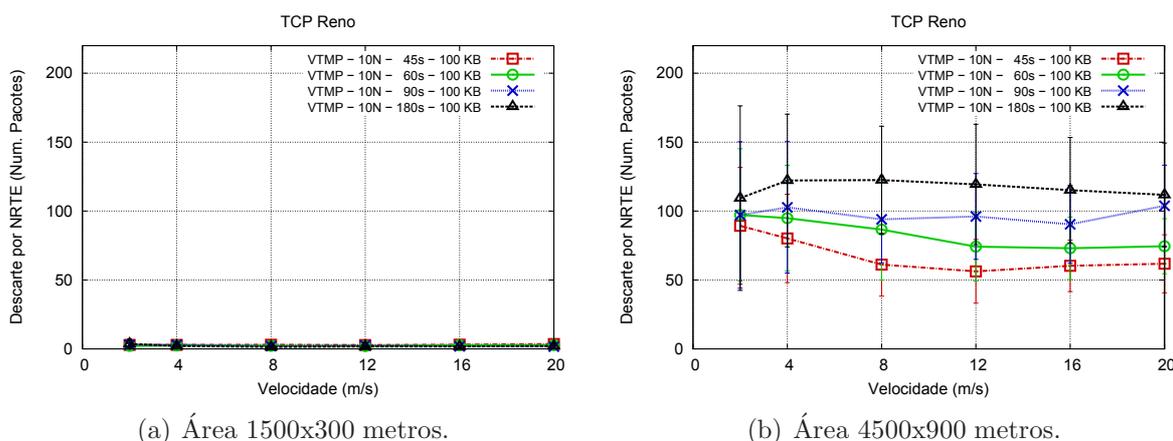


Figura 5.9: Descarte de pacotes por falta de rota considerando arquivos de 100 KB.

Na área de 4500x900 metros (Figura 5.9(b)), o tempo de atualização que apresentou menor número de descartes foi 45s e o tempo que apresentou maior número de descartes foi 180s. A diferença entre os dois tempos foi de 20 pacotes e 50 pacotes para uma velocidade de 2m/s e 20m/s respectivamente.

Na transmissão de arquivos de 800 KB em uma área de 4500x900 metros (Figura 5.10(b)), o tempo de atualização de rotas que apresentou menor número de descartes foi o tempo de 45s e o que apresentou maior número de pacotes descartados por falta de rota foi o tempo de 180s. O tempo de 45s descartou 302 pacotes e 194 pacotes a menos que o tempo de 180s, isso para a

velocidade de movimentação dos nós de 2m/s e 20m/s respectivamente.

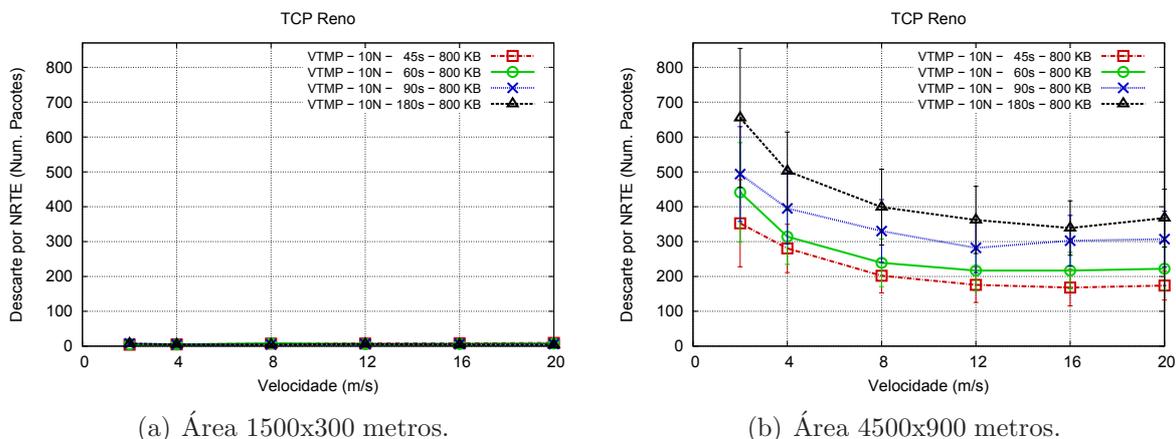


Figura 5.10: Descarte de pacotes por falta de rota considerando arquivos de 800 KB.

Na métrica de descartes de pacotes por falta de rota, o tempo de atualização que apresentou melhor resultado foi o tempo de 45s, considerando uma rede esparsa. Nas simulações com densidade alta não há diferença significativa entre os tempos de atualização de rota. Observa-se menor descartes por falta de rota no tempo de atualização de 45s, isso ocorre pois com atualizações mais frequentes mantém-se mais rotas ativas. O descarte por falta de rota é inexistente na área densa, pois mantém-se caminhos mais curtos entre uma origem e um destino, o que garante rotas mais confiáveis.

Nas Figuras 5.11 e 5.12 são apresentados os resultados de sobrecarga de pacotes de controle, para arquivos de 100 KB e 800 KB respectivamente.

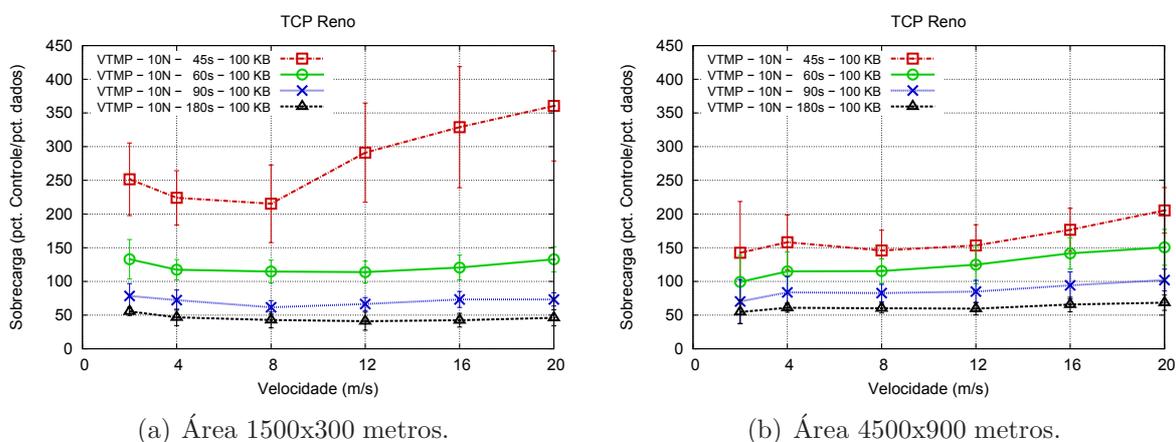


Figura 5.11: Sobrecarga de pacotes de controle considerando arquivos de 100 KB.

No cenário de maior densidade de nós (Figura 5.11(a)), o tempo de atualização que apre-

sentou menor sobrecarga foi 180s, e o tempo que apresentou maior sobrecarga foi o tempo de 45s. A diferença entre esses tempos de atualização é estatisticamente significativa, sendo que é de 196 pacotes de controle/pacotes de dados (PC/PD) para uma movimentação dos nós a 2m/s e 315 PC/PD para uma movimentação dos nós a 20m/s.

Nas Figuras 5.11(b) e 5.12(b) são apresentados os resultados dos cenário com a rede esparsa. Em ambos os casos, o tempo de atualização de 180s obteve a menor sobrecarga, e o tempo de 45s apresentou maior sobrecarga. Os tempos de 60s e 90s seguem uma escala de aumento linear conforme diminui o tempo aumenta-se a sobrecarga.

Na Figura 5.12(a) pode-se observar, assim como nos demais cenários, que o tempo de atualização de 180s obteve menor sobrecarga, e o tempo que apresentou maior sobrecarga foi 45s. A diferença entre esses tempos de atualização é estatisticamente significativa, neste caso o tempo de 45s teve uma sobrecarga de 55 PC/PD a mais, para uma movimentação dos nós a 2m/s e 75 PC/PD a mais para uma movimentação dos nós a 20m/s.

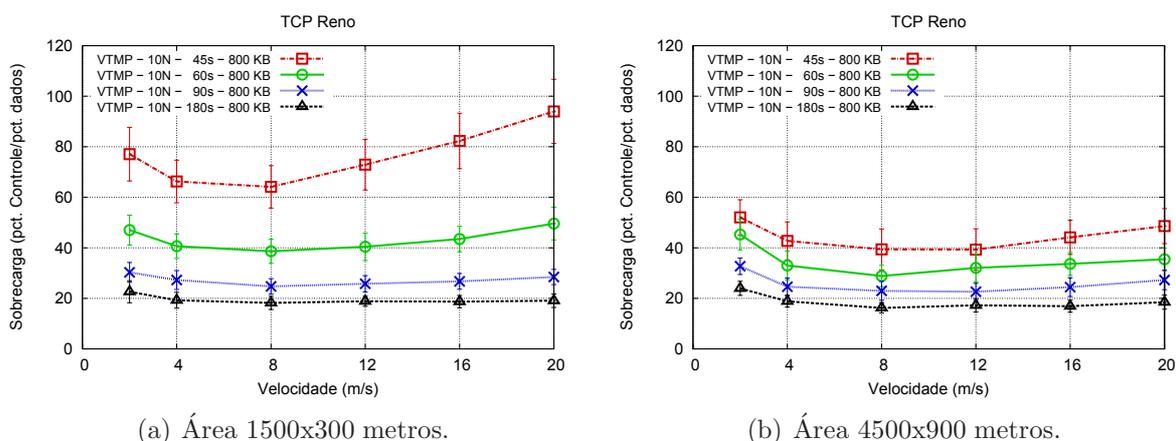


Figura 5.12: Sobrecarga de pacotes de controle considerando arquivos de 800 KB.

Na métrica de sobrecarga, como já era esperado, o tempo de atualização de rotas que apresentou a menor sobrecarga foi o tempo de 180s e o tempo que apresentou maior sobrecarga foi o tempo de 45s. Isso se deve ao fato de quanto menor o tempo entre as atualizações de rotas mais pacotes de controle serão gerados aumentando a sobrecarga de pacotes de controle durante o funcionamento da rede.

Os resultados demonstraram que nas métricas de atraso e descartes de pacotes por falta de rota os tempos de 45s e 60s apresentaram melhores resultados. Isso ocorre, pois com atualizações mais frequentes há mais rotas ativas e atualizadas, em contrapartida há um aumento significativo

nos pacotes de controle, o que degrada o desempenho efetivo da rede.

Nas métricas taxa de entrega, vazão efetiva, descarte total de pacotes e sobrecarga os tempos de 90s e 180s apresentaram melhores resultados. Com uma sobrecarga menor há uma melhora nos resultados de algumas métricas, quando comparado com os tempos de 45s e 60s. Os resultados dos tempos de 90s e 180s são muito próximos e em alguns casos sem uma diferença significativa ao considerar o desvio padrão.

O tempo de atualização entre as rotas do VTMP que foi definido para as demais simulações foi o tempo de 90s, principalmente por ter obtido resultados melhores que os tempos de 45s e 60s na maioria das métricas, resultados iguais ou muito próximos do tempo de 180s e levando em consideração as características do protocolo AODV, utilizado nas simulações.

Os protocolos de roteamento reativos mantêm as rotas criadas por um tempo pré-determinado ou enquanto estiverem sendo utilizadas. Isso aumenta a chance da rota ser válida quando requisitada. Caso uma rota não esteja registrada na tabela de roteamento o protocolo inicia novamente o processo de descoberta de rota. Como a proposta do VTMP é de manter as rotas ativas e prontas para serem utilizadas pelo TCP, o tempo de atualização de rota deve ser o mais próximo possível do tempo de validade das rotas mantidas pelo protocolo de roteamento, de modo a não comprometer o funcionamento da rede. Neste caso, como estamos utilizando o protocolo AODV, entre os tempos de 90s e 180s, que obtiveram melhores resultados, foi definido o tempo de 90s para as atualizações das rotas virtuais.

## 5.4 Considerações Finais

Neste capítulo definiu-se os parâmetros utilizados para as simulações de avaliação do VTMP. A avaliação experimental apresentada neste capítulo foi realizada para definir o tempo de atualização das rotas do VTMP. Por meio dessas simulações foi definido o tempo de atualização ideal, para garantir menor impacto na rede mantendo a qualidade dos serviços.

No capítulo 6, são apresentados os resultados das simulações com e sem o uso do VTMP sob diferentes implementações do TCP, avaliando seu desempenho sob cada uma das implementações.

## CAPÍTULO 6

### RESULTADOS DA AVALIAÇÃO EXPERIMENTAL DO PROTOCOLO VTMP

Neste capítulo são apresentadas as avaliações do impacto do VTMP sobre a MANET utilizando diferentes implementações do protocolo TCP. Na Tabela 6.1 estão definidos os parâmetros específicos utilizados nestas simulações, sendo que os parâmetros gerais foram definidos na Seção 5.2.

Tabela 6.1: Cenários de simulação do TCP.

Parâmetros	Valores
Número de nós	128
Área	4500x900 m e 1500x300 m
Protocolo de roteamento	AODV
Protocolo de Transporte	TCP Reno, TCP Tahoe, TCP Vegas e TCP New Jersey
Tamanho dos arquivos	100 e 800 KB
Número de conexões simultâneas	10 e 20
Intervalos de atualização de rota	90 segundos

Cada gráfico contém quatro cenários simulados, sendo que cada linha representa um cenário. As marcações verticais de cada ponto no gráfico são o desvio padrão da métrica avaliada, resultante de 35 simulações para cada ponto.

Os cenários presentes em cada gráfico são: A avaliação do funcionamento normal da MANET, apenas com o protocolo de roteamento AODV em uma área com alta densidade de nós (área de 1500x300 m) e outro cenário com os nós esparsos (área de 4500x900 m); Outro cenário é com o uso do protocolo VTMP, também em uma área com alta densidade de nós e outra com os nós esparsos.

Os gráficos estão dispostos em pares para facilitar a visualização do impacto do aumento no tráfego de dados na rede. Esses gráficos estão na mesma escala, já os gráficos que apresentam os resultados de transmissão de arquivos com tamanhos diferentes, 100 KB e 800 KB, em algumas métricas não estão na mesma escala.

Os resultados das simulações com as diferentes implementações do TCP são apresentados

nas seções a seguir. Na seção 6.1 são apresentados os resultados com o uso do TCP Reno; na seção 6.2 são apresentados os resultados com o uso do TCP Tahoe; na seção 6.3 são apresentados os resultados com o uso do TCP Vegas; e na seção 6.4 são apresentados os resultados com o uso do TCP New Jersey.

## 6.1 Resultados do protocolo VTMP sob o TCP Reno

Nesta seção é apresentada a avaliação do VTMP sob a implementação do TCP Reno.

Na Figura 6.1 é apresentado o atraso na entrega de pacotes de dados, em um cenário com envio de arquivos de 100 KB. No gráfico pode-se observar que o VTMP adiciona atraso na transmissão de dados, principalmente em uma área densa. Com 10 conexões simultâneas o aumento médio no atraso foi de 367ms com os nós movendo-se a 2m/s e com os nós movendo-se a 20m/s a diferença foi de 193ms. Com 20 conexões simultâneas os resultados são muito similares, pode-se observar ainda que com baixa velocidade de movimentação dos nós há uma aumento no atraso com o uso do VTMP. Isso ocorre pois o VTMP, devido as traduções de rotas virtuais para rotas físicas, cria caminhos mais longos para alcançar o destino de um pacote, enquanto que sem o uso do VTMP neste cenário denso, o AODV consegue entregar o pacote com atraso menor.

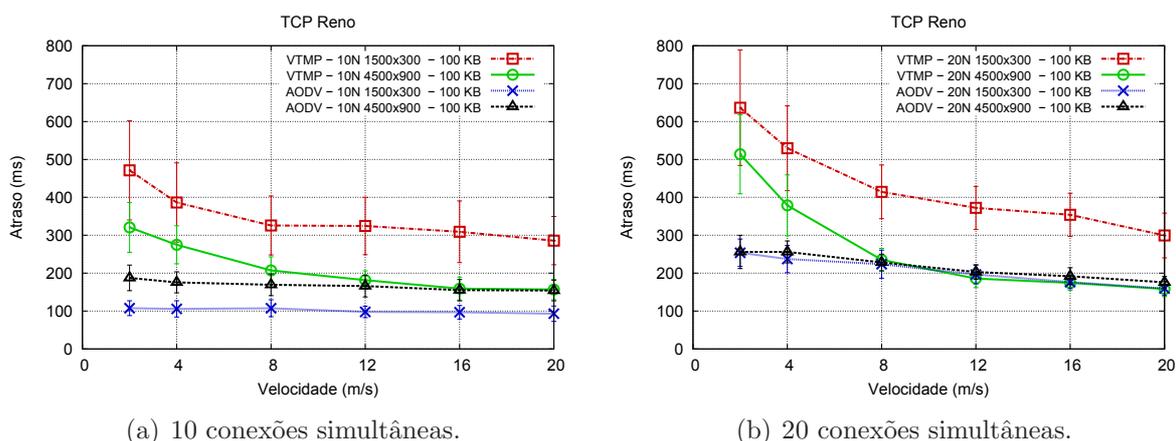


Figura 6.1: Atraso considerando arquivos de 100 KB.

Nos cenários com a rede esparsa e 10 conexões simultâneas, o uso do VTMP aumenta o atraso dos pacotes, enquanto a velocidade de movimentação dos nós é baixa. Neste caso, quanto a velocidade de movimentação dos nós é de 2m/s, o atraso com o uso do VTMP é 133ms maior do que o mesmo cenário sem o uso do VTMP. Com o aumento da velocidade de movimentação

dos nós neste cenário, o atraso com o uso do VTMP se aproxima do atraso na rede sem o uso do VTMP. Quando a velocidade de movimentação dos nós é superior a 8m/s não há diferença significativa no atraso. Os resultados nos cenários com 20 conexões TCP simultâneas são semelhantes ao com 10 conexões simultâneas.

No cenário com arquivos maiores sendo transmitidos, arquivos de 800 KB, apresentados na Figura 6.2, os resultados são similares ao cenário com arquivos de 100 KB. Com a rede densa o VTMP aumenta o atraso na entrega de pacotes, pode-se observar na Figura 6.2(a) que com os nós movendo-se a 2m/s o atraso adicionado é de 346ms e a 20m/s o atraso é de 159ms. Quando há mais conexões simultâneas a diferença no atraso provocado pelo VTMP diminui, apresentado um aumento no atraso de 188ms a uma velocidade de 2m/s e 100ms a uma velocidade de 20m/s.

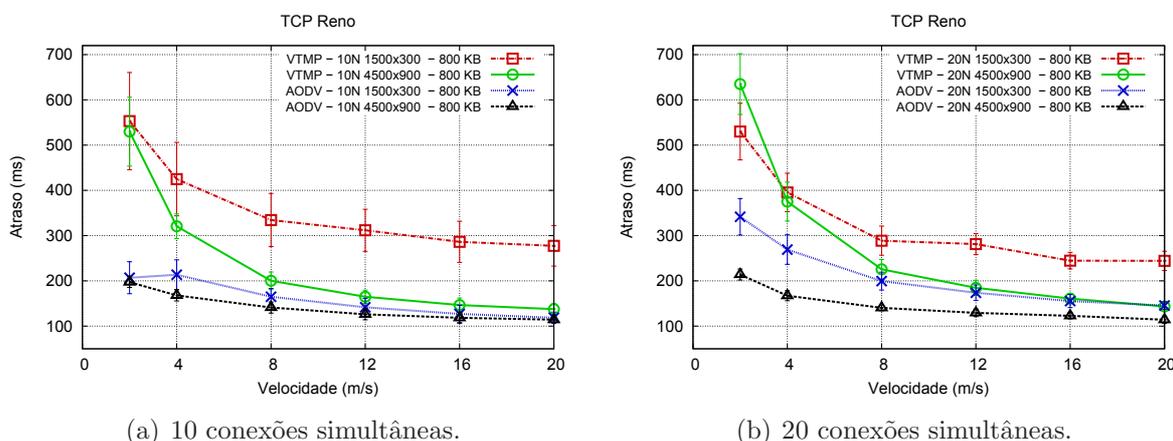


Figura 6.2: Atraso considerando arquivos de 800 KB.

O atraso com o uso do VTMP aumenta na rede esparsa e transmissão de arquivos maiores, Figura 6.2. Com os nós movendo-se a 2m/s o atraso adicional é de 332ms, em um cenário com 10 nós transmitindo simultaneamente, e 420ms em um cenário com 20 nós transmitindo. Nestes cenários quando a velocidade dos nós é superior a 8m/s a diferença do atraso diminui consideravelmente, chegando a aumentar apenas 23ms com o uso do VTMP a uma velocidade de 20m/s.

O atraso adicionado pelo uso do VTMP é ocasionado por dois motivos: rotas maiores e sobrecarga. O VTMP envia os dados por meio de rotas virtuais, uma ligação virtual pode ser traduzida em várias ligações reais, o que aumenta o tempo que o pacote fica na rede. A manutenção das rotas virtuais do VTMP gera pacotes de controle para a criação das rotas virtuais na rede real, sendo que os pacotes de controle competem com os pacotes de dados na

camada de roteamento, aumentando assim o tempo que o pacote fica na fila do roteamento.

Na Figura 6.3 são apresentados os resultados da taxa de entrega para arquivos de 100 KB. No cenário de alta densidade de nós observa-se que o uso do VTMP causou uma queda na taxa de entrega de 4%. Essa queda ocorreu nos dois cenários, com 10 e 20 conexões simultâneas. Já no cenário com a rede esparsa, o uso do protocolo VTMP melhorou a taxa de entrega, aumentando em média 1%, em relação a rede sem o VTMP, mas ao considerar o desvio padrão não há diferença significativa entre os resultados neste cenário. Esse resultado manteve-se com o aumento do número de conexões simultâneas, de 10 conexões para 20 conexões.

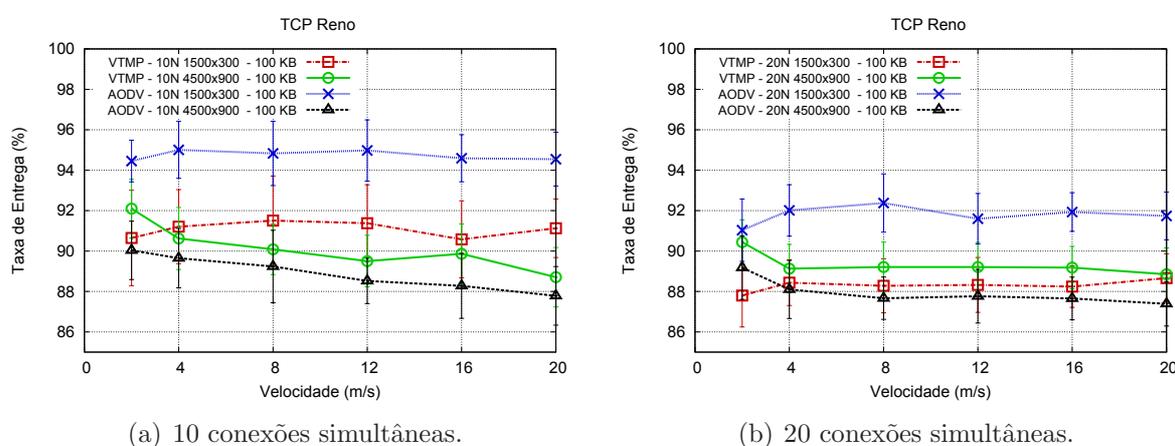


Figura 6.3: Taxa de Entrega considerando arquivos de 100 KB.

Na Figura 6.4 é apresentada a taxa de entrega da transmissão de arquivos de 800 KB. No cenário de alta densidade, observa-se que quando os nós estão movendo-se a 2m/s o uso do VTMP não influencia a taxa de entrega. Com velocidade igual ou superior a 4m/s o VTMP causa uma perda de 1% na taxa de entrega, em comparação com a rede sem o VTMP, no cenário com 10 conexões simultâneas. No cenário com 20 conexões simultâneas, o VTMP só influencia na taxa de entrega quando a velocidade de movimentação dos nós é igual ou superior a 12m/s. Neste caso, ele diminui a taxa de entrega em até 2%.

Considerando o cenário com a rede esparsa e 10 conexões simultânea, apresentado na Figura 6.4, o uso do VTMP melhora a taxa de entrega em 4% com os nós movendo-se a 2m/s, e 1% com os nós movendo-se a 20m/s. Neste mesmo cenário, mas com 20 conexões simultâneas, o uso do VTMP melhorou a taxa de entrega em 6%, com os nós movendo-se a 2m/s, e 2% com os nós movendo-se a 20m/s

Na rede densa no geral o VTMP diminui a taxa de entrega. Um dos motivos é o menor

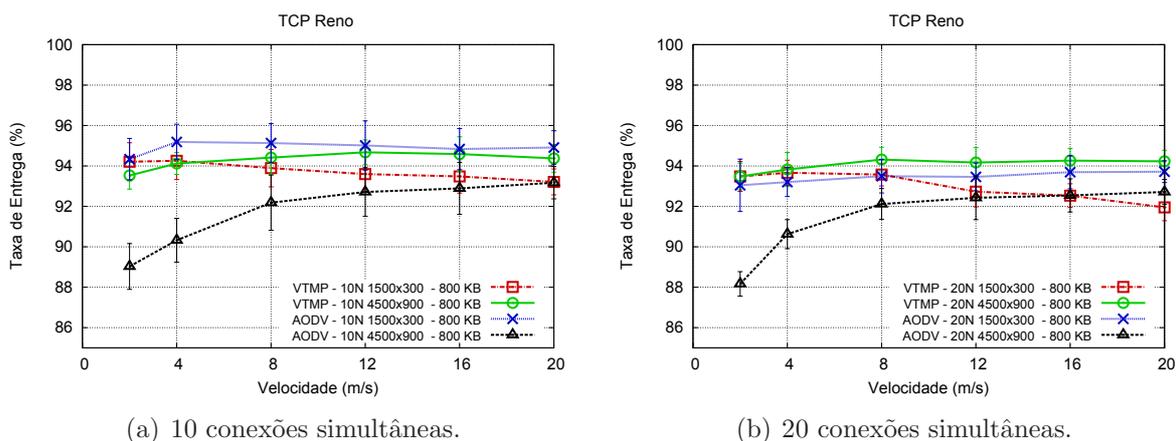


Figura 6.4: Taxa de Entrega considerando arquivos de 800 KB.

tamanho das rotas quando não é utilizado o VTMP, como a rede é densa o número de saltos para entrega de um pacote com o uso do VTMP se torna maior, e mesmo mantendo mais rotas ativas o VTMP pode perder pacotes por quebra de rota e sobrecarga. Na rede esparsa ocorre o contrário, há um aumento na taxa de entrega, podendo ser explicado pelo fato da rede esparsa obrigar o AODV a criar rotas mais longas, enquanto que a rede com o VTMP já possui as rotas ativas enviando diretamente os pacotes.

Na métrica vazão efetiva é onde o VTMP apresenta maior impacto negativo. O cenário com transmissão de arquivos de 100 KB é apresentado na Figura 6.5. Pode-se observar que no cenário com 10 conexões simultâneas e rede densa, o uso do VTMP diminuiu em média 950 B/s da vazão efetiva. No cenário com 20 conexões simultâneas o uso do VTMP diminuiu em média 1100 B/s da vazão efetiva.

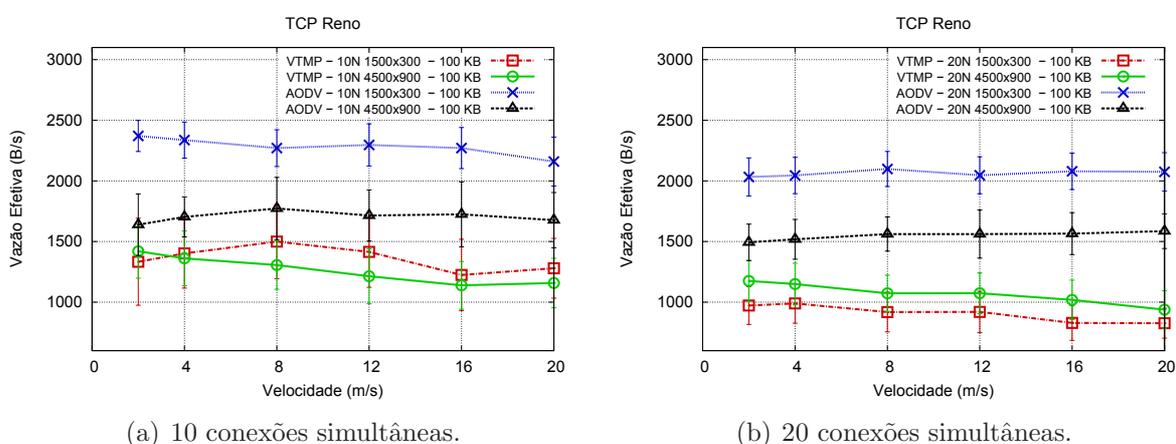


Figura 6.5: Vazão Efetiva considerando arquivos de 100 KB.

No cenário com a rede esparsa o impacto é menor, com 10 conexões simultâneas a vazão é 220 B/s menor usando o VTMP com os nós movendo-se a 2m/s e 520 B/s menor com os nós movendo-se a 20m/s. Na simulação de 20 conexões simultâneas o uso do VTMP representou uma redução na vazão de 320 B/s e 647 B/s com os nós movendo-se a 2m/s e 20m/s, respectivamente. Observa-se uma redução na vazão efetiva com o aumento do número de conexões simultâneas, que ocorre pois há um aumento de colisões e uso da banda por diferentes conexões, como o meio é compartilhado a vazão efetiva é afetada.

Os cenários com transmissão de arquivos de 800 KB são apresentados na Figura 6.6. Nas simulações com densidade alta de nós e uso do VTMP, observa-se que conforme aumenta a velocidade de movimentação dos nós diminui a vazão efetiva. No cenário com 10 conexões simultâneas o uso do VTMP na rede diminui a vazão em 3408 B/s com os nós movendo-se a 2m/s, e 5450 B/s com os nós movendo-se a 20m/s. No cenário com 20 conexões simultâneas o resultado apresenta uma curva de decréscimo da vazão muito próxima a curva que foi apresentada com 10 conexões simultâneas.

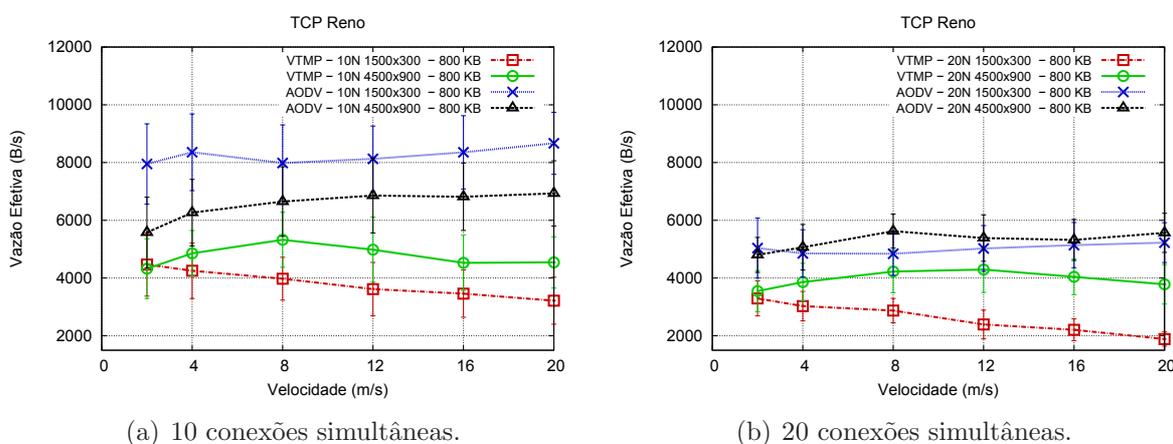


Figura 6.6: Vazão Efetiva considerando arquivos de 800 KB.

No cenário com uma rede esparsa (Figura 6.6) o uso do VTMP também diminui a vazão efetiva. Observa-se que para um cenário com 10 conexões simultâneas e os nós movendo-se a 2m/s a vazão diminuiu 1250 B/s, e com os nós movendo-se a 20m/s a vazão foi 2387 B/s menor. No cenário com 20 conexões simultâneas ocorre uma redução da vazão similar a redução que ocorreu no cenário com 10 conexões.

Na métrica vazão efetiva, devido ao maior tamanho das rotas e maior sobrecarga na rede há uma redução dessa taxa com o uso do VTMP. O melhor cenário considerando essa métrica foi

a rede mais esparsa com arquivos de 800 KB e 20 conexões simultâneas.

Na Figura 6.7 é apresentado o resultado de descarte total de pacotes em cenários com transmissão de arquivos de 100 KB. No cenário de alta densidade de nós e 10 conexões simultâneas, o uso do VTMP aumenta o número de pacotes descartados, em média são descartados 100 pacotes a mais do que no cenário sem o uso do VTMP. No cenário de 20 conexões simultâneas ocorreu um aumento no número de descartes, em média foram descartados 400 pacotes a mais do que no cenário sem o uso do VTMP .

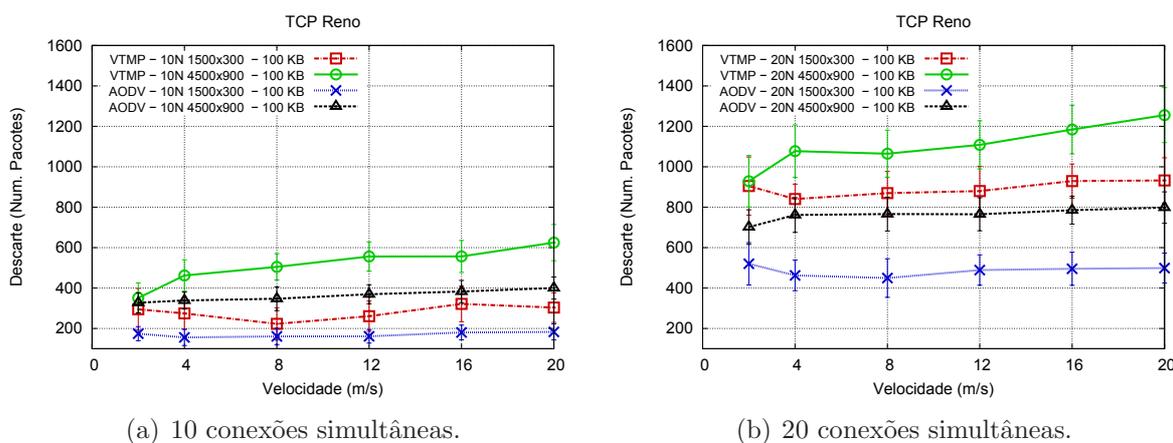
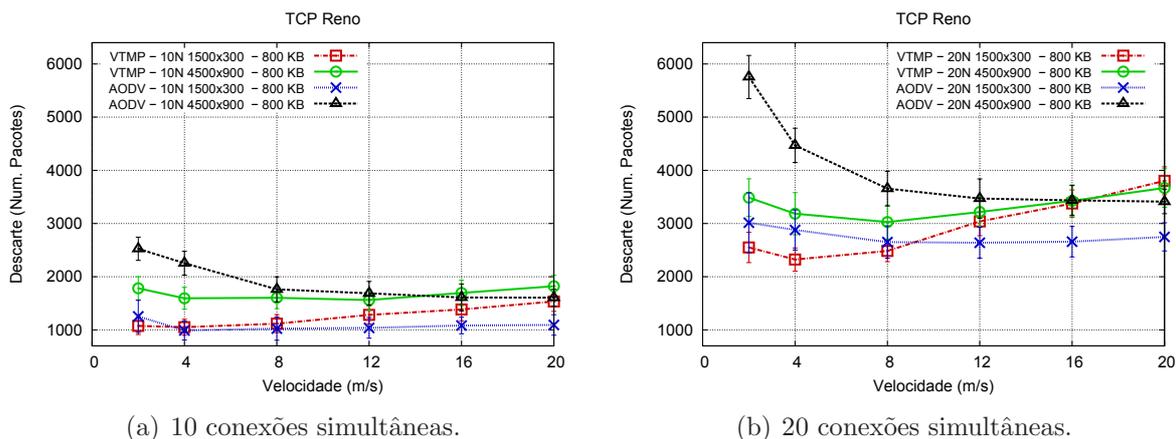


Figura 6.7: Descarte de pacotes considerando arquivos de 100 KB.

No cenário com a rede esparsa e com 10 conexões simultâneas, o uso do VTMP aumenta o número de descartes quanto a velocidade de movimentação dos nós é igual ou superior a 4m/s. Neste caso, o número de descartes aumenta em até 225 pacotes, quando a velocidade de movimentação alcança 20m/s. No cenário com 20 conexões simultâneas, o uso do VTMP também aumentou o número de descartes em relação a rede sem o VTMP, neste caso, o aumento foi de 230 à 460 pacotes.

Na Figura 6.8 é apresentado o resultado de descarte total de pacotes com os nós transmitindo arquivos de 800 KB. Observa-se que com 10 conexões simultânea, no cenário de alta densidade de nós, o uso do VTMP aumenta os descartes de pacotes quando a velocidade de movimentação dos nós é superior a 8m/s. Neste caso, há um aumento nos descartes de até 440 pacotes. No cenário com 20 conexões simultâneas o uso do VTMP reduz os descartes de pacotes enquanto a velocidade dos nós é inferior a 12m/s. Quando a velocidade de movimentação é igual ou superior a 12m/s o uso do VTMP aumenta do descarte total de pacotes, descartando até 1050 pacotes a mais com os nós movendo-se a 20m/s.

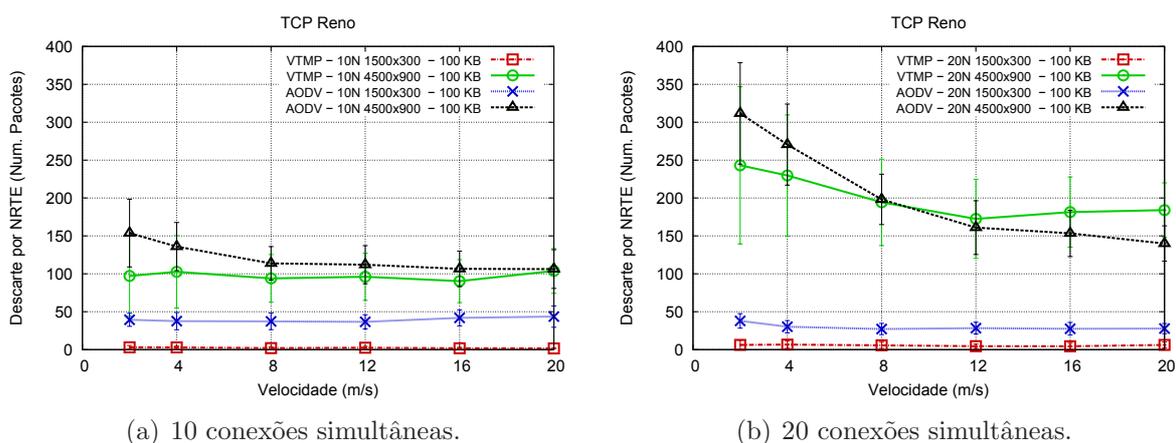


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.8: Descarte de pacotes considerando arquivos de 800 KB.

No cenário da rede esparsa e 10 conexões simultâneas, apresentado na Figura 6.8, o uso do VTMP diminui o descarte de pacotes. Com os nós movendo-se a uma velocidade de até 2m/s, houve uma redução média nos descartes de 740 pacotes, com o uso do VTMP. Quando a velocidade de movimentação dos nós é igual ou superior a 8m/s essa diferença nos descartes passa a não ser significativa, ao considerar o desvio padrão. No cenário com 20 conexões o uso do VTMP também reduz o descarte de pacotes, sendo descartados 2268 pacotes a menos, quando a velocidade de movimentação é de 2m/s. Com o aumento da velocidade de movimentação dos nós essa diferença diminui, passando a não ser significativa a partir do momento que os nós se movem com velocidade igual ou superior a 12 m/s.



(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.9: Descarte de pacotes por falta de rota considerando arquivos de 100 KB.

Na Figura 6.9 é apresentado o resultado do descarte de pacotes por falta de rota, na transmissão de arquivos de 100 KB. Considerando o cenário de alta densidade e com 10 conexões

TCP simultâneas, observa-se que o uso do VTMP reduziu a quase zero o descarte dos pacotes de dados por falta de rota, apresentando uma redução média de 40 pacotes. No cenário com 20 conexões simultâneas o resultado é similar, a redução média no número de descarte ficou em 25 pacotes.

Considerando o cenário esparsos e 10 conexões simultâneas, também ocorreu redução no descarte por falta de rota. Neste cenário obteve-se uma redução de 56 pacotes, no descarte de pacotes por falta de rota sem o uso do VTMP, quando a velocidade de movimentação dos nós é de até 2m/s. A partir da movimentação dos nós a uma velocidade igual ou superior a 4m/s, o uso do VTMP na média reduz o descarte de pacotes, mas considerando o desvio padrão a diferença não é significativa. No cenário com 20 conexões simultâneas o uso do VTMP não altera o total de descartes de pacotes, quando considerado o desvio padrão. Apenas com os nós movendo-se a 20m/s há uma diferença significativa, neste caso o uso do VTMP causou um aumento nos descartes de 45 pacotes a mais, quando comparado com o mesmo cenário sem o uso do VTMP.

A métrica de descarte de pacotes por falta de rota em um cenário com transmissão de arquivos de 800 KB é apresentada na Figura 6.10. No cenário com alta densidade de nós e 10 conexões simultâneas, o uso do protocolo VTMP diminuiu o descarte de pacotes por falta de rota a quase zero, em média reduziu os descartes em 120 pacotes. No mesmo cenário mas com 20 conexões simultâneas a redução no número de descartes se manteve, neste caso descartando em média 240 pacotes a menos do que no cenário sem o uso do VTMP.

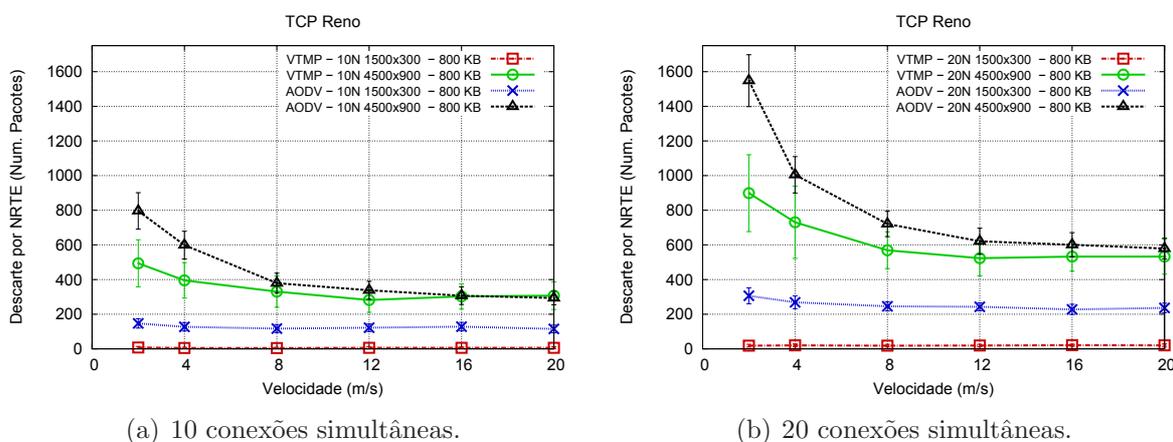


Figura 6.10: Descarte de pacotes por falta de rota considerando arquivos de 800 KB.

No cenário com a rede esparsa e 10 conexões TCP simultâneas, o uso do VTMP diminuiu

o descarte de pacotes em quando a velocidade de movimentação é baixa, descartando até 300 pacotes a menos com os nós movendo-se a 2m/s. Quando a velocidade é igual ou superior a 8m/s, o uso do VTMP não apresenta mais diferença significativa ao se considerar o desvio padrão. No cenário com 20 conexões simultâneas o uso do VTMP reduziu o número de descartes em 649 pacotes, com os nós movendo-se a 2m/s. Conforme aumenta a velocidade dos nós da rede o uso do VTMP fica mais próximo do resultado sem o uso do VTMP, sendo que a partir da velocidade de 12 m/s não há diferença significativa no uso do VTMP.

Na métrica de descarte de pacotes por falta de rota, em geral, há uma redução no número desses descartes, ocasionada pelo fato do VTMP manter rotas atualizadas e ativas. Na rede esparsa o número de descarte por falta de rota é maior quando comparado com os descartes na rede densa, isso ocorre pois na rede esparsa os caminhos são mais longos e as quebras de rotas ocorrem mais frequentemente, o que gera mais descartes.

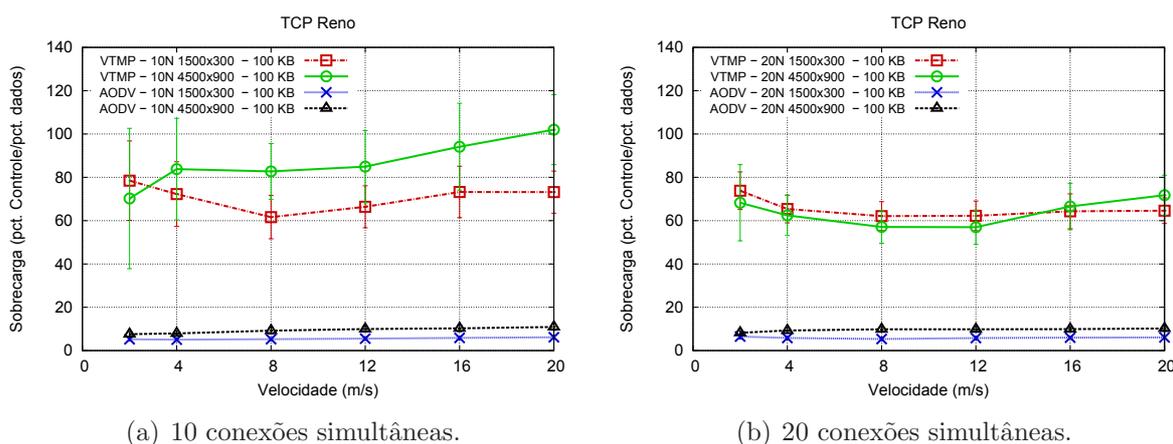
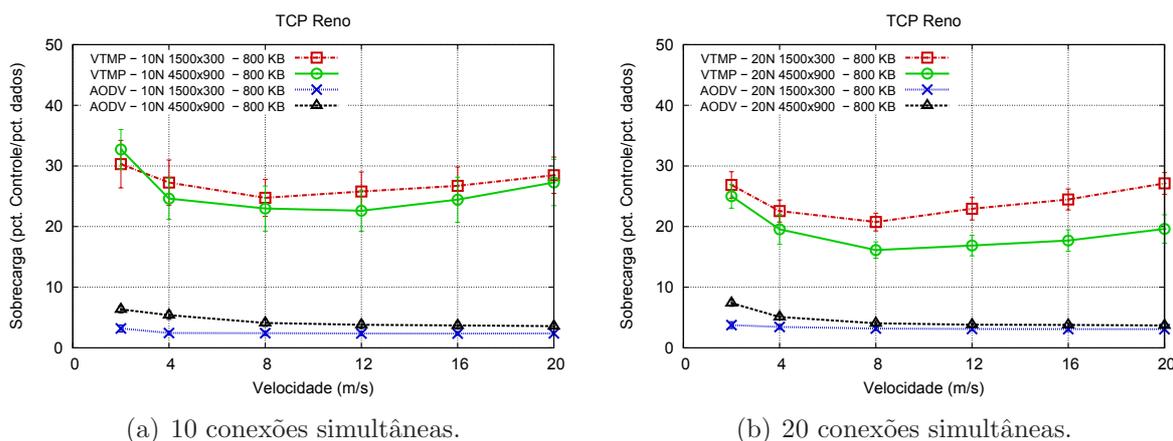


Figura 6.11: Sobrecarga considerando arquivos de 100 KB.

A métrica sobrecarga em cenários com transferências de arquivos de 100 KB é apresentada na Figura 6.11. O impacto na sobrecarga já era esperado, pois independente do envio de dados o VTMP envia pacotes de controle periodicamente para manter as rotas atualizadas. O uso do VTMP em uma rede com alta densidade de nós e 10 conexões TCPs simultâneas, em média aumentou a sobrecarga em 65 PC/PD. No cenário com 20 conexões simultâneas a sobrecarga foi de 59 PC/PD a mais do que na rede sem o uso do VTMP.

Com a rede esparsa e 10 conexões simultâneas, em média o uso do VTMP enviou 76 PC/PD a mais em comparação ao cenário sem o uso do VTMP. Já no cenário com 20 conexões TCP simultâneas foram enviados 54 PC/PD a mais para cada pacote de dado.



(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.12: Sobrecarga considerando arquivos de 800 KB.

A métrica sobrecarga em cenários com transferências de arquivos de 800 KB é apresentada na Figura 6.12. Em um cenário com alta densidade de nós e com 10 conexões simultâneas, a sobrecarga com o uso do VTMP também foi maior. Neste cenário a diferença em relação ao cenário sem o uso do VTMP foi de 24 PC/PD. Já no cenário com 20 conexões simultâneas o uso do VTMP gerou 21 PC/PD a mais.

No cenário esparsa e 10 conexões simultâneas, o uso do VTMP gerou em média 21 PC/PD a mais do que no cenário sem o uso do VTMP. No cenário com 20 conexões simultâneas o uso do VTMP aumentou a sobrecarga em 14 PC/PD.

Por meio dos gráficos pode-se observar que com o aumento do tráfego de dados na rede a sobrecarga gerada pelo VTMP diminui. Isso ocorre pois o VTMP utiliza os pacotes de controle para manter as rotas virtuais ativas, mesmo sem a transmissão de dados, o que gera a sobrecarga maior do protocolo.

Os resultados mostram que o uso do VTMP sob o TCP Reno obteve melhores resultados em um rede esparsa e de baixa velocidade de movimentação dos nós. Isso ocorre pois nesses cenários há menor quebra das rotas, garantindo uma maior conectividade. Em cenários com a rede densa o uso do VTMP degrada o desempenho da MANET, em relação a métrica de taxa de entrega, isso ocorre pelo fato do uso do VTMP gerar rotas maiores para os pacotes, o que aumenta a chance de quebra de rotas e a perda dos pacotes.

Outra característica do uso do VTMP é a redução da vazão efetiva, ocasionada pelas características do TCP, o qual reage negativamente a perdas de pacotes. Com as perdas de pacotes o TCP mantém a janela de congestionamento baixa com o uso do VTMP, com isso a vazão efetiva

é menor.

Os descartes de pacotes com o uso do VTMP foram maiores em alguns dos cenários, principalmente quando a densidade de nós é maior. Isso ocorreu pois o protocolo de roteamento não prioriza os pacotes de dados, desse modo com a rede densa os pacotes de controle do VTMP podem acarretar em filas no protocolo de roteamento, criando gargalos. Os resultados de descarte por falta de rota comprovam esse fato, pois com o uso do VTMP a maior parte dos descartes não foi por falta de rota e sim outros motivos, entre eles erro de bit e filas do roteamento.

## 6.2 Avaliação experimental do protocolo VTMP sob o TCP Tahoe

Nesta seção é apresentada a avaliação do VTMP sob a implementação do TCP Tahoe, para avaliar e validar o funcionamento do protocolo proposto.

Nas Figuras 6.13 e 6.14 são apresentados os resultados da métrica atraso nos cenários com transmissão de arquivos de 100 KB e 800 KB respectivamente. Em todos os cenários os resultados são semelhantes aos resultados obtidos com a implementação do TCP Reno, apresentados na Seção 6.

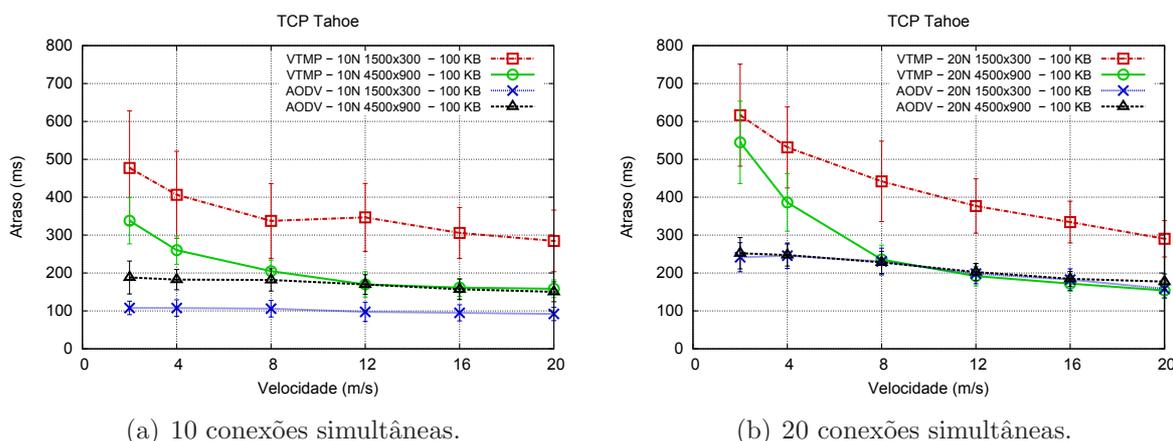
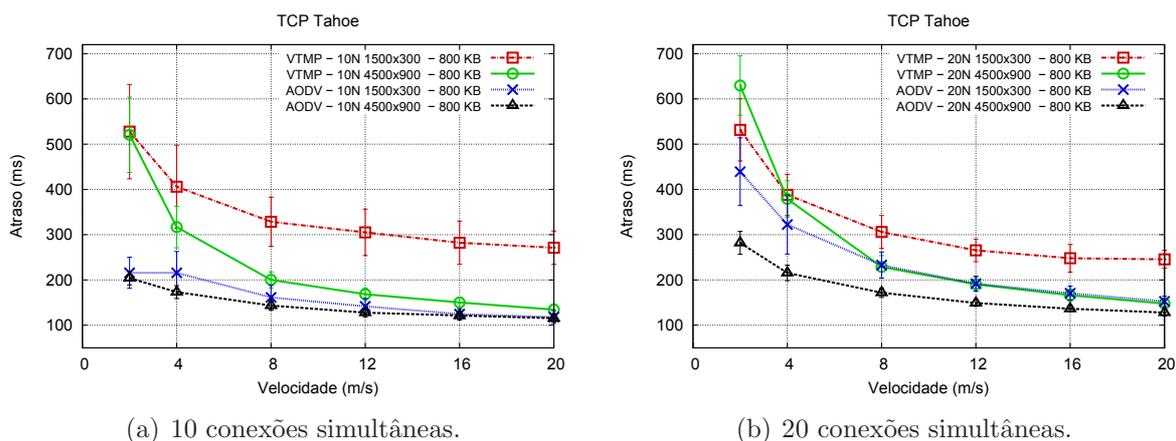


Figura 6.13: Atraso considerando arquivos de 100 KB.

No cenário com a rede esparsa e com o uso do VTMP, o atraso aumenta quando a velocidade de movimentação dos nós é baixa. Isso ocorre pois neste cenário as conexões virtuais criadas pelo VTMP se mantêm por mais tempo, forçando muitas vezes os pacotes a seguir por um caminho mais longo até o destino, quando comparado com o mesmo cenário sem o uso do VTMP. Quando



(a) 10 conexões simultâneas.

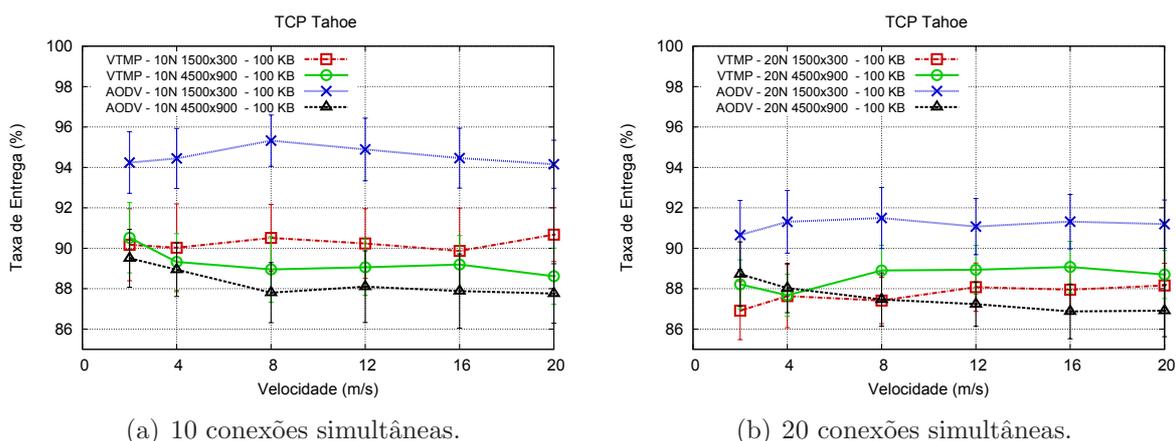
(b) 20 conexões simultâneas.

Figura 6.14: Atraso considerando arquivos de 800 KB.

a velocidade de movimentação dos nós aumenta ambas as rotas, na rede com e sem o uso do VTMP, quebram-se mais rapidamente, reduzindo assim a diferença do atraso. Com velocidade de movimentação nós igual ou superior 8m/s não há mais diferença significativa no atraso.

No cenário com a rede densa, a maior proximidade dos nós e o menor número de saltos entre dois nós da rede garantem um atraso menor para a rede sem o uso do VTMP.

Nas Figuras 6.15 e 6.16 são apresentados os resultados da taxa de entrega para a transmissão de arquivos de 100 KB e 800 KB. Os resultados com o TCP Tahoe foram semelhantes aos resultados apresentados com o TCP Reno.

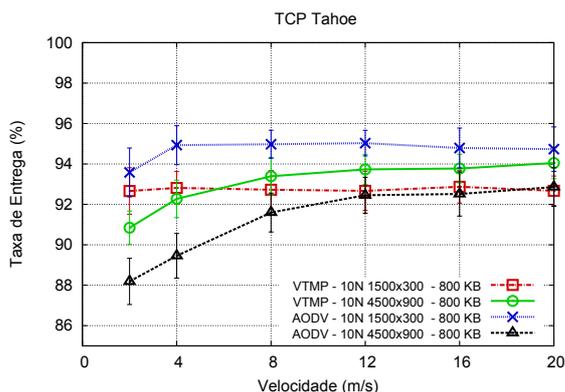


(a) 10 conexões simultâneas.

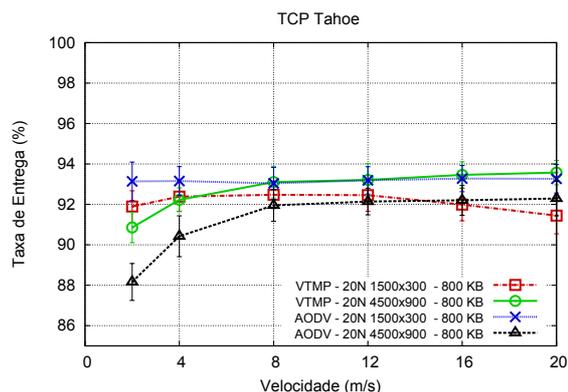
(b) 20 conexões simultâneas.

Figura 6.15: Taxa de Entrega considerando arquivos de 100 KB.

A taxa de entrega na rede densa, de 1500x300 metros, foi reduzida com o uso do VTMP. Isso ocorreu pois na rede sem o VTMP as rotas são mais curtas, o que permite uma maior quantidade de pacotes de dados serem enviados quando a rota é estabelecida.



(a) 10 conexões simultâneas.

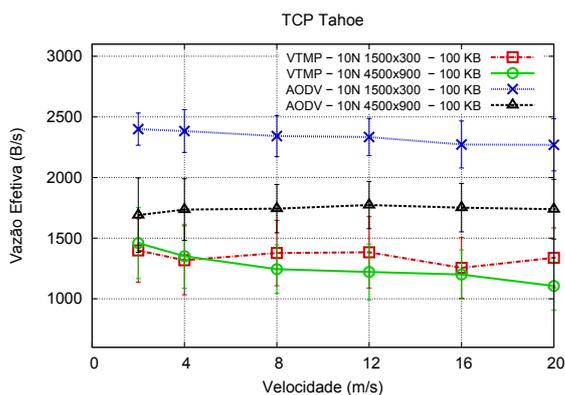


(b) 20 conexões simultâneas.

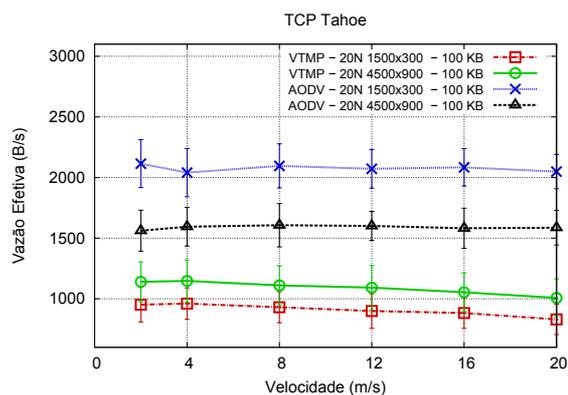
Figura 6.16: Taxa de Entrega considerando arquivos de 800 KB.

Na rede esparsa de 4500x900 metros, o uso do VTMP, na média, melhorou a taxa de entrega. O aumento da taxa de entrega neste cenário ocorre pois a conectividade é garantida pelo uso do VTMP e o cenário sem o VTMP não tem mais a vantagem dos caminhos mais curtos que a rede densa propicia.

Os resultados obtidos na métrica vazão efetiva com o TCP Tahoe são similares aos resultados obtidos com o TCP Reno, havendo uma redução na taxa de envio de dados com o uso do VTMP. Os resultados da vazão efetiva com transmissão de arquivos de 100 e 800 KB são apresentados nas Figuras 6.17 e 6.18.



(a) 10 conexões simultâneas.



(b) 20 conexões simultâneas.

Figura 6.17: Vazão Efetiva considerando arquivos de 100 KB.

O uso do VTMP diminuiu a vazão efetiva em ambos os cenários, na rede densa e na rede esparsa. A redução da vazão efetiva ocorre por causa do aumento na sobrecarga da rede e pelo maior número de descartes de pacotes, o que gera uma redução na janela de congestionamento

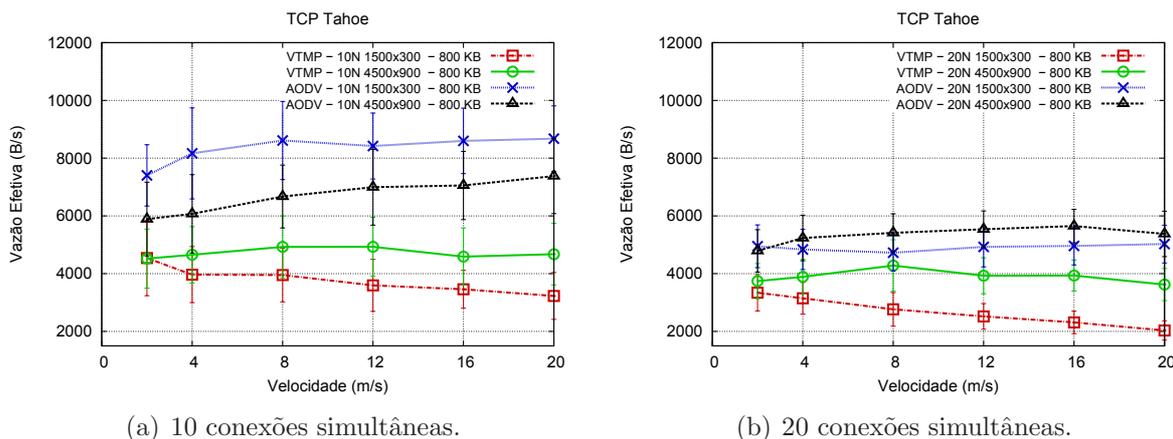


Figura 6.18: Vazão Efetiva considerando arquivos de 800 KB.

do TCP e conseqüentemente a redução da vazão efetiva.

A métrica descarte total de pacotes é apresentada nas Figuras 6.19 e 6.20 com transmissão de arquivos de 100 e 800 KB respectivamente.

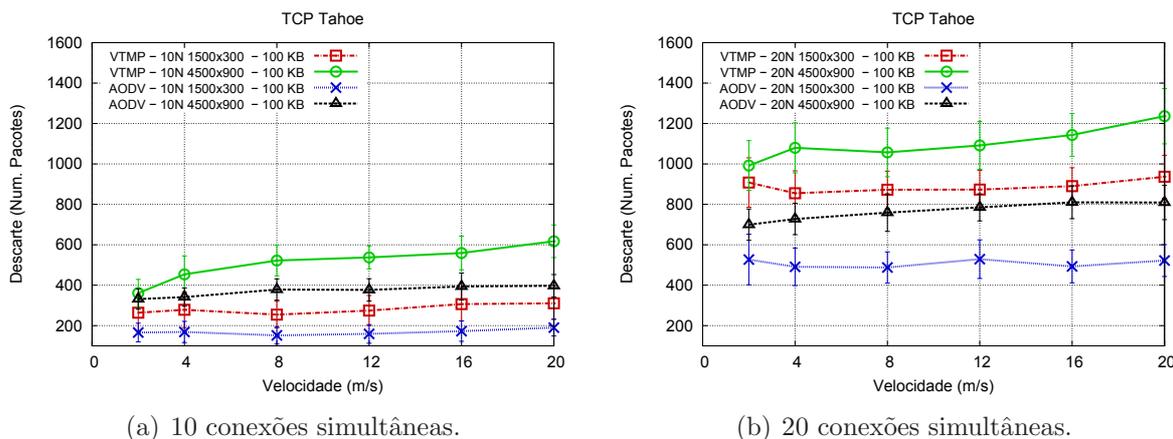
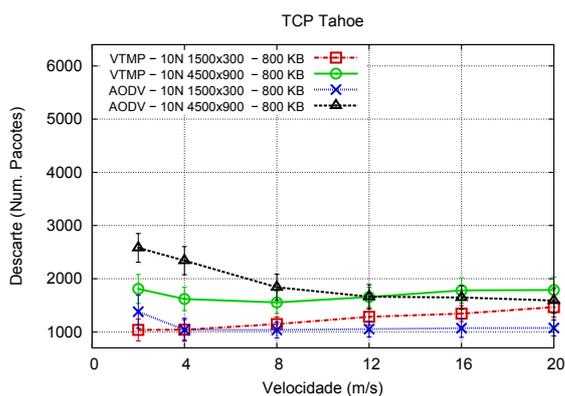


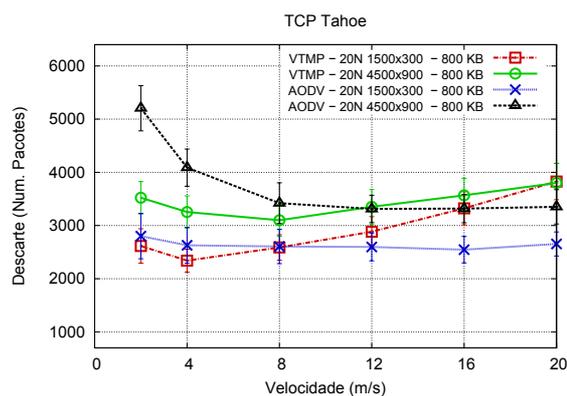
Figura 6.19: Descarte de pacotes considerando arquivos de 100 KB.

No cenário com transmissão de arquivos pequenos de 100 KB, o descarte aumentou com o uso do VTMP na rede densa e na rede esparsa. O aumento no número de descartes pode ser atribuído a sobrecarga gerada pelo VTMP, que compete em nível de roteamento com os pacotes de dados, podendo descartar pacotes por estouro nas filas e estouro de tempo limite do pacote.

No cenário com transmissão de arquivos maiores, de 800 KB, o descarte não é tão afetado pelo uso do VTMP. Isso ocorre pois com arquivos maiores o número de pacotes a ser enviado é maior e conseqüentemente há um aumento no número de descartes. A rede com o uso do VTMP não é tão afetada, neste caso, pois sua sobrecarga reduz a janela de congestionamento do TCP



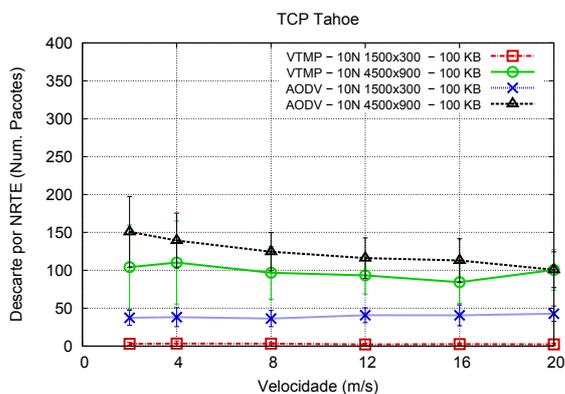
(a) 10 conexões simultâneas.



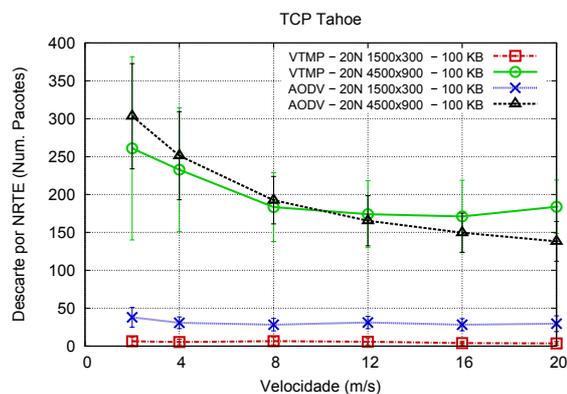
(b) 20 conexões simultâneas.

Figura 6.20: Descarte de pacotes considerando arquivos de 800 KB.

e consequentemente menos pacotes são enviados simultaneamente, reduzindo os descartes.

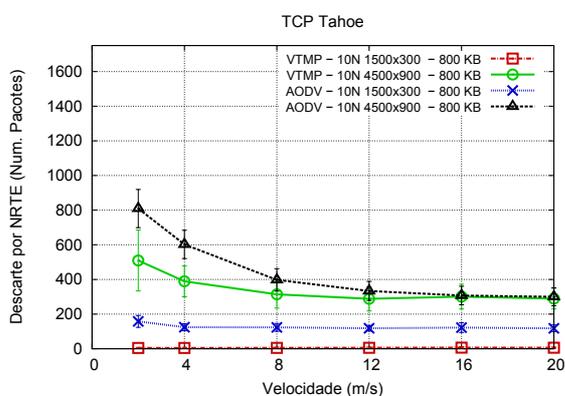


(a) 10 conexões simultâneas.

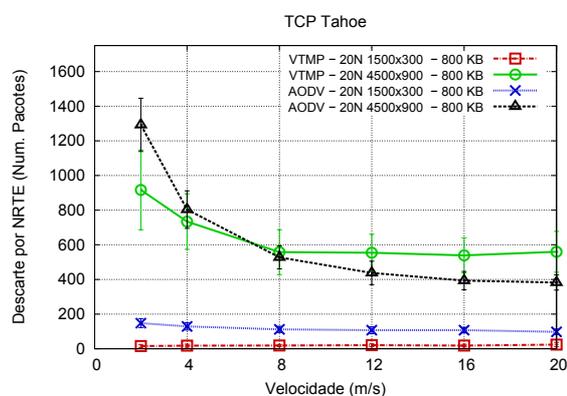


(b) 20 conexões simultâneas.

Figura 6.21: Descarte de pacotes por falta de rota considerando arquivos de 100 KB.



(a) 10 conexões simultâneas.



(b) 20 conexões simultâneas.

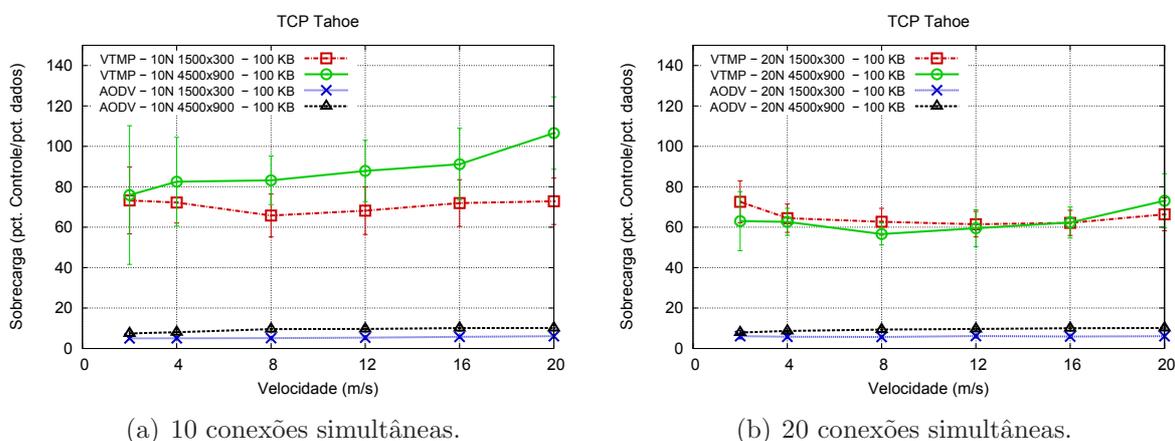
Figura 6.22: Descarte de pacotes por falta de rota considerando arquivos de 800 KB.

Nas Figuras 6.21 e 6.22 são apresentados os resultados dos descartes por falta de rota com

arquivos de 100 e 800 KB.

O descarte por falta de rota é reduzido com o uso do VTMP em redes densas, pois no geral as rotas mantidas pelo VTMP auxiliam na entrega dos pacotes por meio de rotas ativas. Nas rede mais esparsas com baixa velocidade de movimentação dos nós, o uso do VTMP reduz os descartes por falta de rota, mas com o aumento na velocidade de movimentação dos nós há uma redução na confiabilidade dos caminho mantidos pelo VTMP o que aumenta o número de rotas quebradas, gerando mais perdas de pacotes por falta de rota.

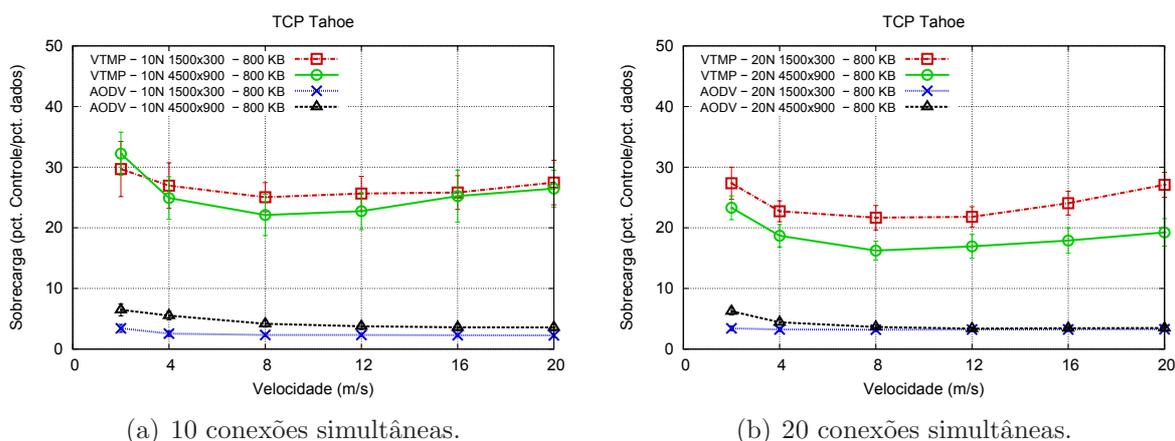
Os resultados da métrica sobrecarga em cenários com transferências de arquivos de 100 e 800 KB são apresentados nas Figuras 6.23 e 6.24.



(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.23: Sobrecarga considerando arquivos de 100 KB.



(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.24: Sobrecarga considerando arquivos de 800 KB.

Na métrica sobrecarga já era esperado um aumento em relação a rede sem o uso do VTMP, pois independente do envio dos dados a rede do VTMP mantém as rotas virtuais pró-ativamente.

Esse processo de atualização de rotas, aumenta o número de pacotes de controle que circulam pela rede, gerando uma sobrecarga. Deste modo quanto maior a quantidade de dados transmitidos menor é a taxa de sobrecarga com o uso do VTMP.

O TCP Tahoe apresentou melhores resultados em redes esparsas e em baixa velocidade de movimentação dos nós. Isso ocorre pois nestes cenários há menor quebra de rotas, o que favorece o uso das rotas virtuais criadas pelo VTMP. Em geral os resultados do uso do VTMP sob o TCP Tahoe são semelhantes aos resultados do TCP Reno, podendo ser utilizada qualquer uma destas implementações do TCP.

### 6.3 Resultados do protocolo VTMP sob o TCP Vegas

Nesta seção é apresentada a avaliação do VTMP sob a implementação Vegas do TCP, a qual é uma abordagem do TCP diferente das apresentadas anteriormente. O TCP Vegas é reativo ao congestionamento da rede, tentando identificar o congestionamento antes que ele aconteça.

Nas Figuras 6.25 e 6.26 são apresentados os resultados do atraso na transmissão de arquivos de 100 e 800 KB.

O TCP Vegas, na média, apresenta uma redução no atraso em comparação as demais implementações do TCP, aqui apresentadas. Essa redução se deve as suas características de ser reativo ao congestionamento e seus algoritmos de controle e reação a perda de pacotes.

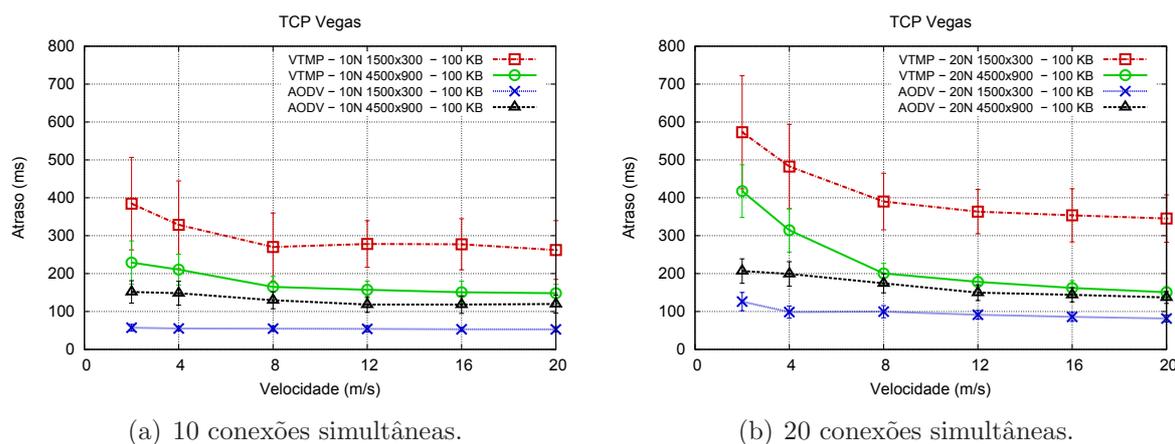
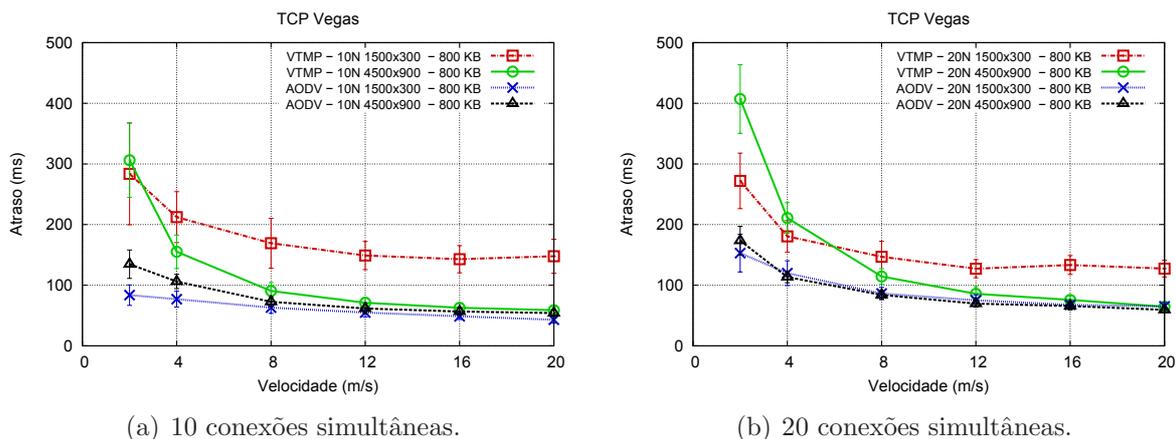


Figura 6.25: Atraso considerando arquivos de 100 KB.

Os resultados do cenário com o uso do VTMP mostram um aumento significativamente no atraso quando a densidade de nós é grande. Esse aumento se deve ao fato do caminho que um pacote percorre na estrutura virtual, na maioria dos casos, será maior que o caminho construído



(a) 10 conexões simultâneas.

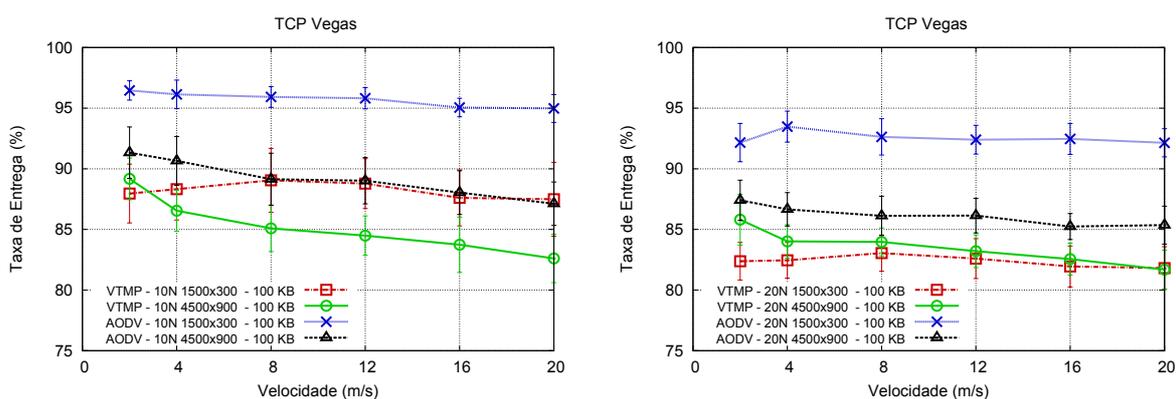
(b) 20 conexões simultâneas.

Figura 6.26: Atraso considerando arquivos de 800 KB.

pele AODV. Outro ponto que influencia neste cenário é a proximidade dos nós, o que mantém as rotas criadas inicialmente por muito mais tempo, não havendo a criação de novas rotas até que as antigas se tornem inválidas, mesmo que as novas rotas fossem mais curtas.

Na rede esparsa as rotas podem ser mais longas, mas com a movimentação dos nós ocorrem mais quebras de rotas, com isso, rotas mais curtas são geradas sempre que necessário. Como pode-se observar nos gráficos o atraso na rede esparsa é menor que na rede densa com o uso do VTMP.

A métrica taxa de entrega em cenários com transmissão de arquivos de 100 e 800 KB são apresentados nas Figuras 6.27 e 6.28 respectivamente.

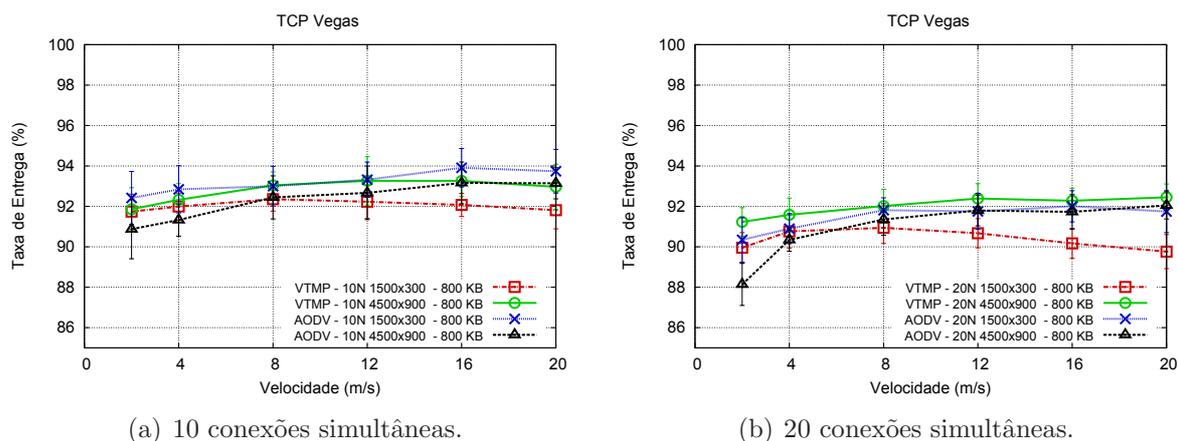


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.27: Taxa de Entrega considerando arquivos de 100 KB.

O uso do VTMP no envio de arquivos pequenos reduz a taxa de entrega na rede densa e na rede esparsa. A redução, nestes casos, ocorreu pelas características do TCP Vegas que



(a) 10 conexões simultâneas.

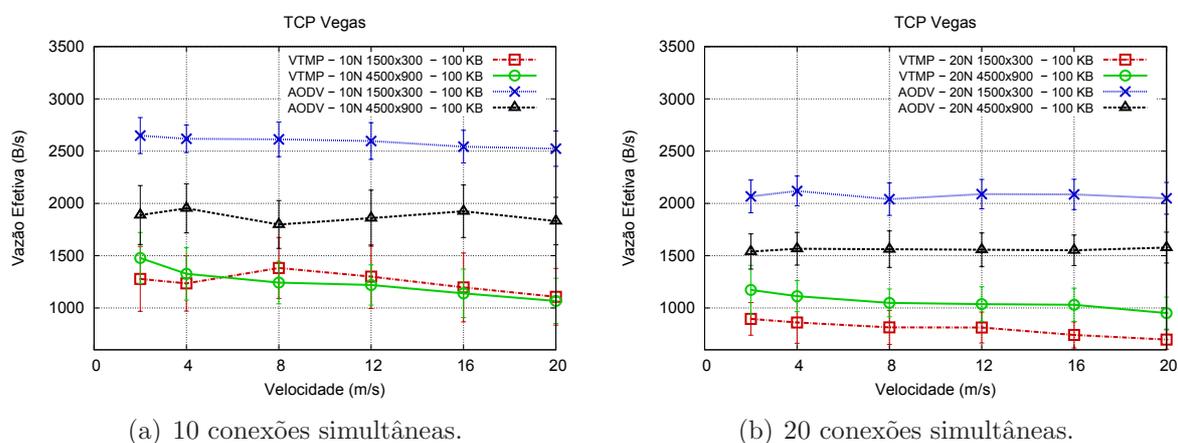
(b) 20 conexões simultâneas.

Figura 6.28: Taxa de Entrega considerando arquivos de 800 KB.

reenvia um pacote mesmo antes de se receber a confirmação de entrega ou mesmo antes do tempo do pacote expirar, para garantir uma recuperação mais rápida de possíveis perdas de pacotes. Levando em consideração que o uso do VTMP gera caminhos mais longos e que a sobrecarga da rede é maior com seu uso, isso pode afetar negativamente o funcionamento dos algoritmos do Vegas, piorando seu desempenho.

A diferença apresentada na taxa de entrega com o uso do VTMP diminui quando os arquivos a serem transmitidos são maiores. Com arquivos maiores há mais pacotes de dados sendo transmitidos simultaneamente na rede, o que degrada o desempenho no cenário sem o uso do VTMP, principalmente na rede esparsa. Neste cenário, na média o uso do VTMP aumenta a taxa de entrega, mas ao considerar o desvio padrão a diferença não é significativa.

Os cenários com os resultados da métrica vazão efetiva são apresentados nas Figura 6.29 e 6.30, com transmissão de arquivos de 100 e 800 KB respectivamente.



(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.29: Vazão Efetiva considerando arquivos de 100 KB.

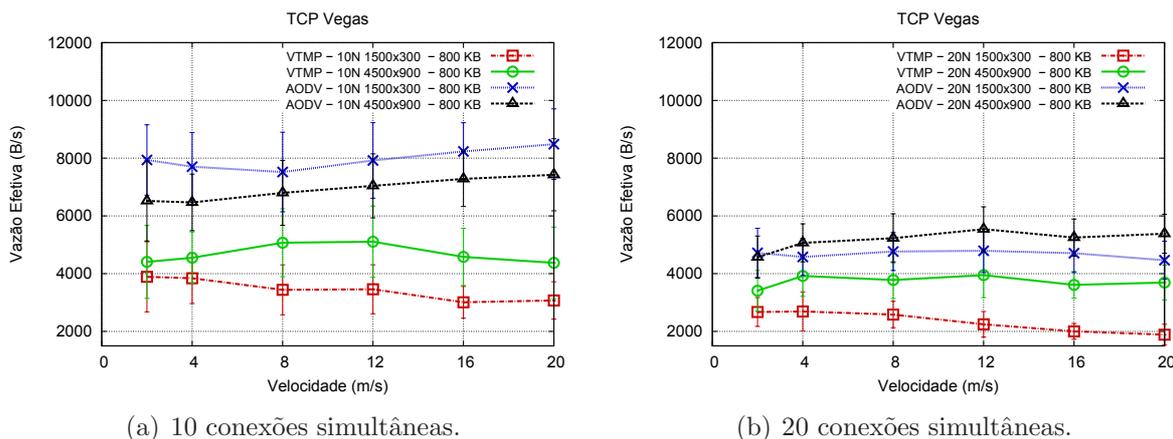


Figura 6.30: Vazão Efetiva considerando arquivos de 800 KB.

Em todos os cenários avaliados o uso do VTMP reduz a vazão efetiva, quando comparado com os cenários sem o uso do VTMP. A redução da vazão ocorreu principalmente por dois motivos: A influência da sobrecarga de pacotes de controle na rede, os quais competem pelo uso da rede com os pacotes de dados; e pelas características do controle de congestionamento do TCP, que reage negativamente a perda de pacotes.

Nas Figuras 6.31 e 6.32 são apresentados os resultados do descarte total de pacotes de dados nos cenários com transmissão de arquivos de 100 KB e 800 KB respectivamente.

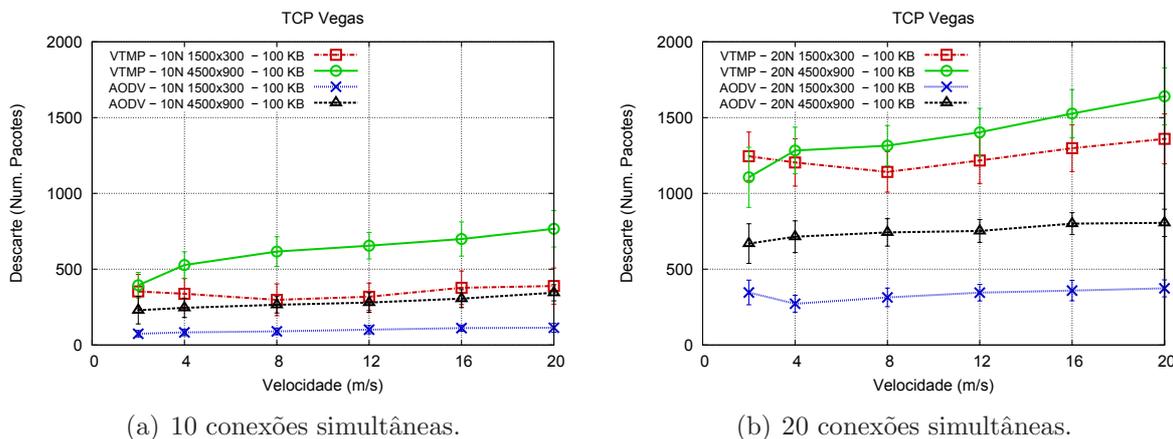
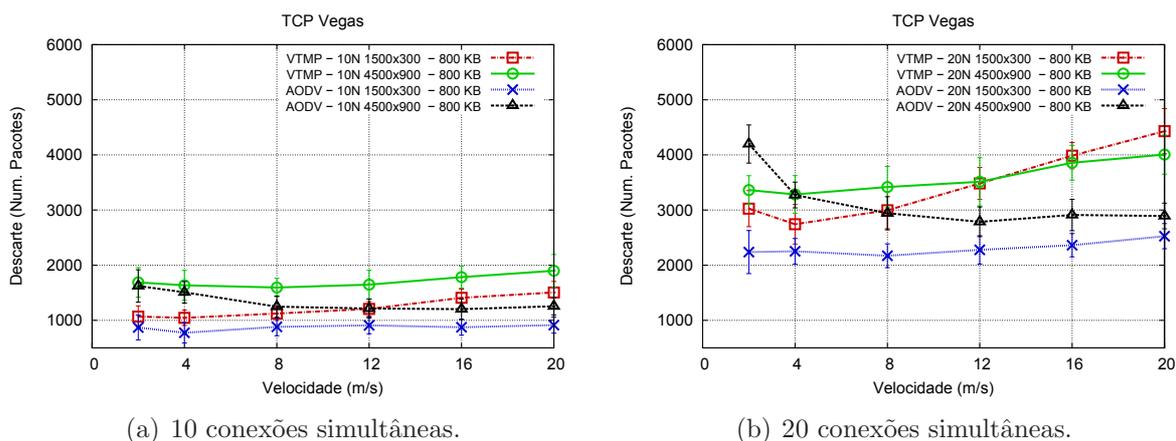


Figura 6.31: Descarte de pacotes considerando arquivos de 100 KB.

O descarte total de pacotes de dados aumenta com o uso do VTMP, principalmente em cenários com transmissão de arquivos pequenos, de 100 KB. Com o aumento do tamanho dos arquivos transmitidos, de 100 para 800 KB, o VTMP ainda apresenta um aumento dos descartes, quando comparado com a rede sem o uso do VTMP, mas a diferença no número total de descartes



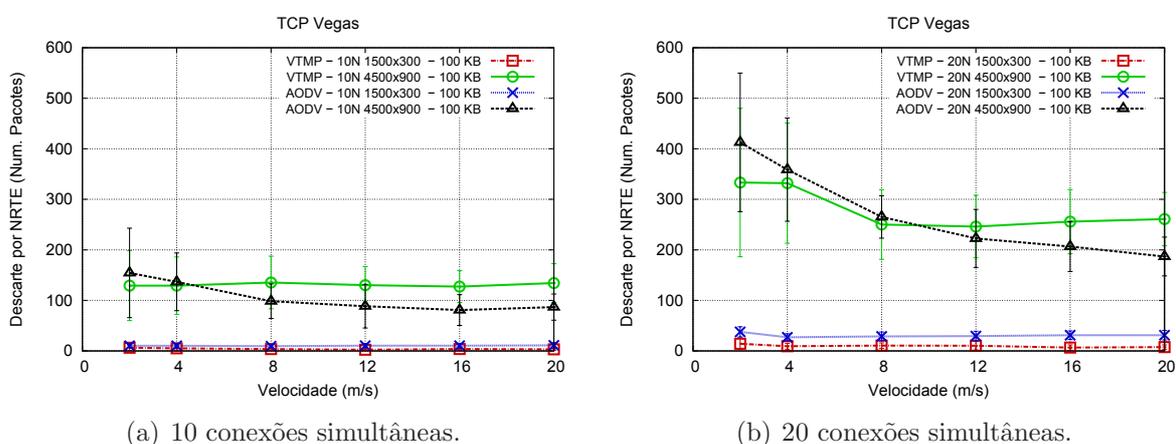
(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.32: Descarte de pacotes considerando arquivos de 800 KB.

reduz. Isso ocorre pois a sobrecarga gerada pelo VTMP aumenta o número de descartes de pacotes de dados, visto que os protocolos de roteamento não priorizam pacotes de dados. Isso pode ser confirmado pelo aumento no número de descartes da rede sem o VTMP com transmissão de arquivos maiores.

Os resultados de descarte total de pacotes em cenários com transmissão de arquivos de 100 e 800 KB são apresentados nas Figuras 6.33 e 6.34 respectivamente.

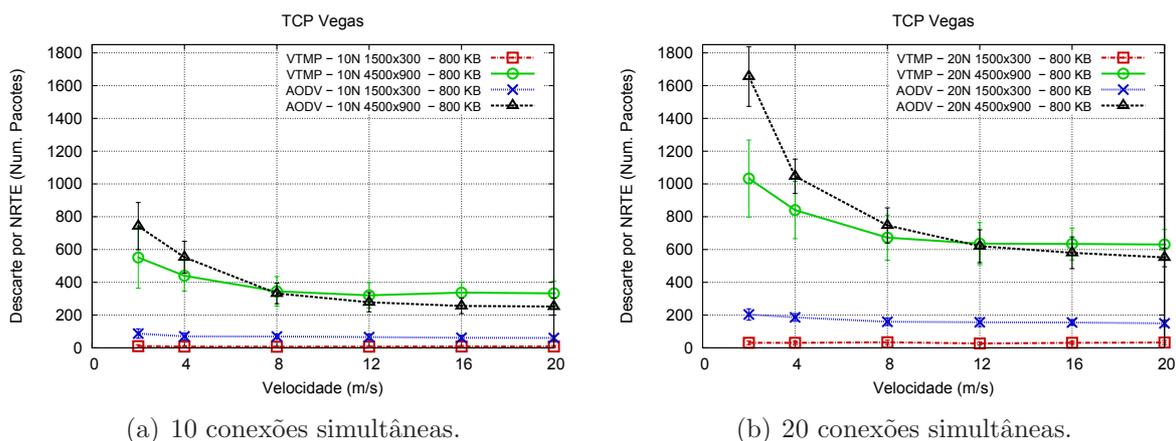


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.33: Descarte de pacotes por falta de rota considerando arquivos de 100 KB.

O uso do VTMP reduz os descartes por falta de rota na rede densa, isso ocorre pois neste cenário as rotas criadas pelo VTMP são mantidas por mais tempo e as rotas são recriadas mais facilmente. No cenário com a rede esparsa não há diferença significativa no uso do VTMP, ele não reduz nem aumenta os descartes por falta de rota no cenário da transmissão de arquivos de 100 KB. No cenário com transmissão de arquivos de 800 KB e rede esparsa, o uso do VTMP



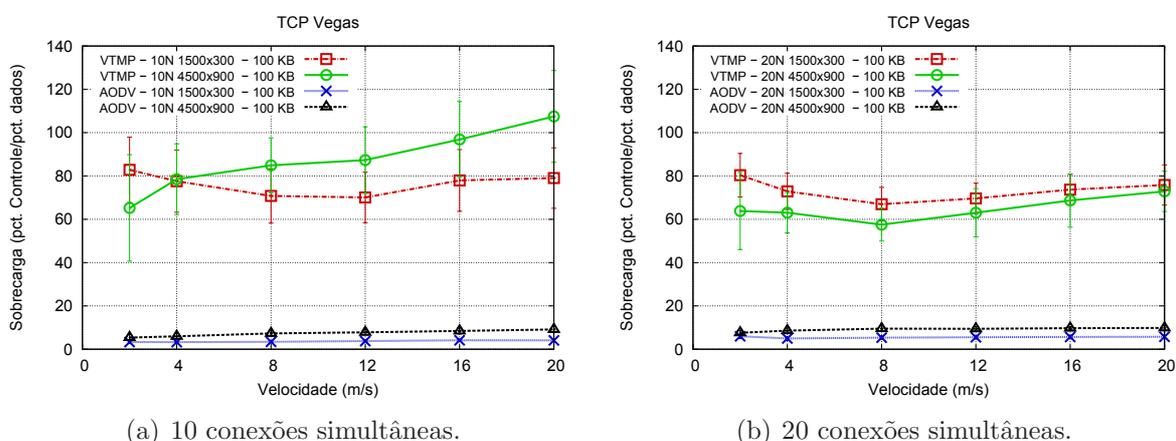
(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.34: Descarte de pacotes por falta de rota considerando arquivos de 800 KB.

reduz os descartes por falta de rota quando a movimentação dos nós é inferior a 8 m/s. Quando a velocidade de movimentação dos nós é igual ou superior a 8m/s a diferença nos descartes não é significativa ao considerar o desvio padrão.

A sobrecarga em cenários com transmissão de arquivos de 100 e 800 KB é apresentada nas Figuras 6.35 e 6.36 respectivamente.



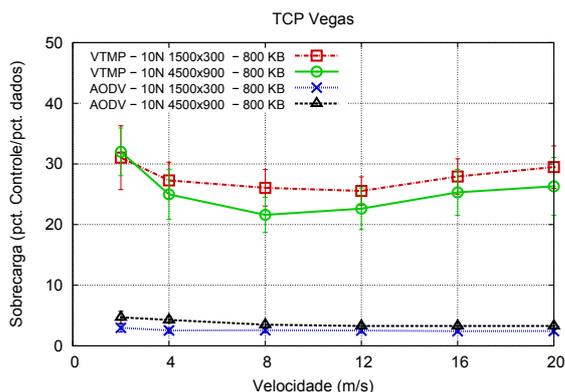
(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

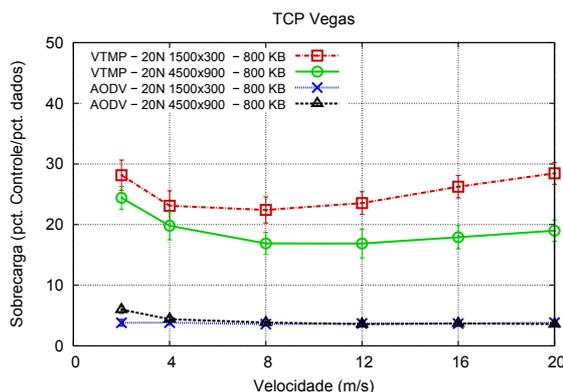
Figura 6.35: Sobrecarga considerando arquivos de 100 KB.

Nos cenários com uso do VTMP há um aumento na sobrecarga, em comparação aos cenários sem o uso do VTMP. O uso do TCP Vegas não influenciou esta taxa, mantendo-se praticamente a mesma das demais avaliações.

Nos resultados do uso do VTMP sob a implementação Vegas do protocolo TCP, as métricas que apresentaram maior diferença em relação as outras implementações do TCP foram o atraso e a taxa de entrega. O TCP Vegas reduz o atraso, devido suas características, mas que conse-



(a) 10 conexões simultâneas.



(b) 20 conexões simultâneas.

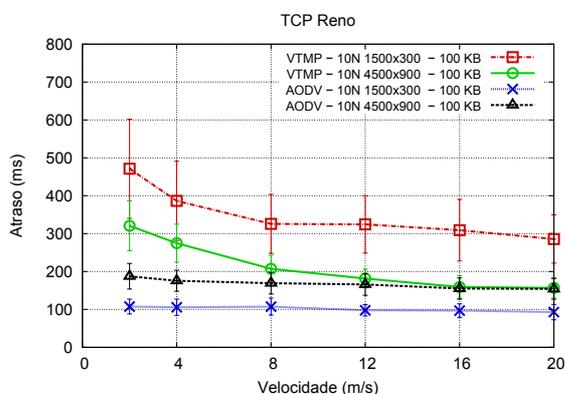
Figura 6.36: Sobrecarga considerando arquivos de 800 KB.

quentemente reduz a taxa de entrega nos cenários com o uso do VTMP. Nas demais métricas não observou-se diferença significativa entre o uso das diferentes implementações do TCP.

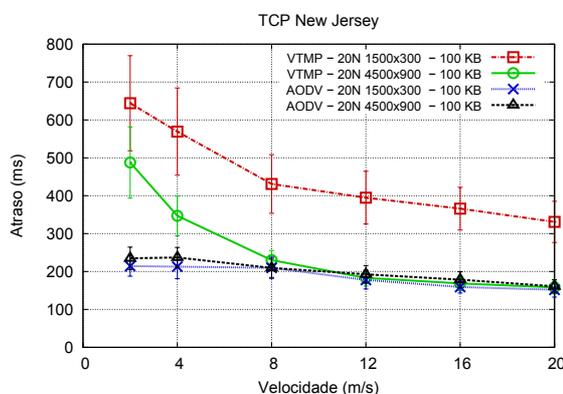
O uso do VTMP sob a implementação do TCP Vegas em alguns cenários não se mostraram a melhor opção. Em cenários com alta densidade de nós e baixa carga de dados o uso do VTMP influência negativamente em todas as métricas. O VTMP combinado com o TCP Vegas apresentou melhores resultados nos cenários com a rede esparsa.

## 6.4 Resultados do protocolo VTMP sob o TCP New Jersey

Nesta seção é apresentada a avaliação do impacto do VTMP sob o uso da implementação New Jersey do TCP, diferentemente das demais implementações, o TCP New Jersey é uma variação do TCP desenvolvida para redes sem fio.



(a) 10 conexões simultâneas.



(b) 20 conexões simultâneas.

Figura 6.37: Atraso considerando arquivos de 100 KB.

Os resultados da métrica atraso para os cenários de transmissão de arquivos de 100 e 800 KB são apresentados nas Figuras 6.37 e 6.38 respectivamente.

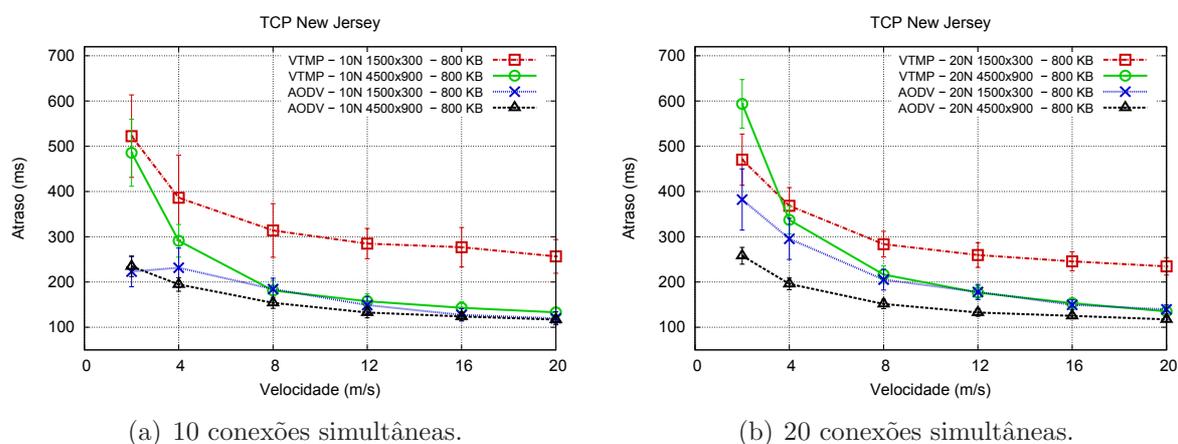


Figura 6.38: Atraso considerando arquivos de 800 KB.

Nos cenários com o uso do TCP NJ o atraso apresentou resultados similares aos resultados do TCP Reno. Na área com alta densidade de nós o uso do VTMP aumenta o atraso, sem apresentar diferença no uso da variante do TCP para redes sem fio. Na área em que os nós estão esparsos o atraso é maior com o uso do VTMP quando a velocidade de movimentação dos nós é baixa. Neste cenário com a velocidade de movimentação dos nós igual ou superior a 8m/s não há diferença significativa no atraso, ao considerar o desvio padrão.

Nas Figuras 6.39 e 6.40 são apresentados os resultados da taxa de entrega nos cenários com transmissão de arquivos de 100 e 800 KB.

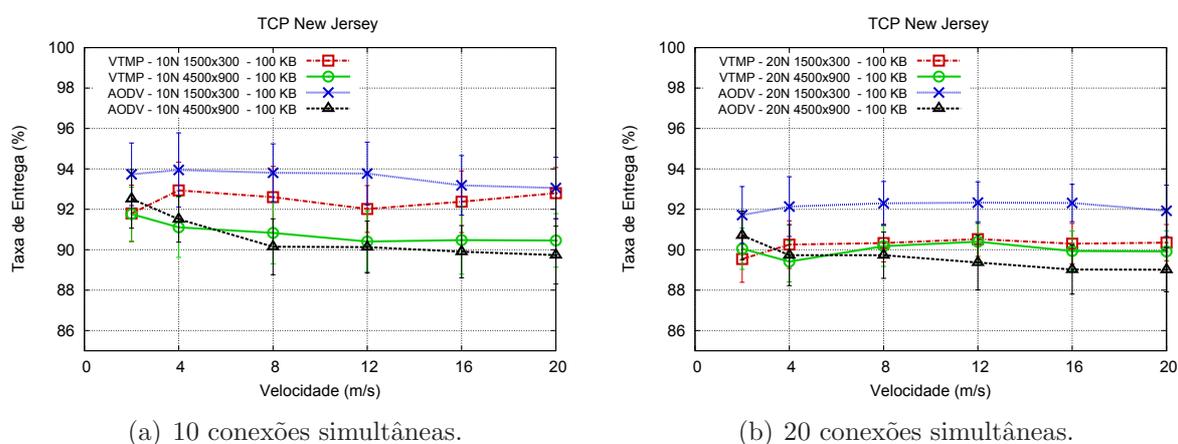
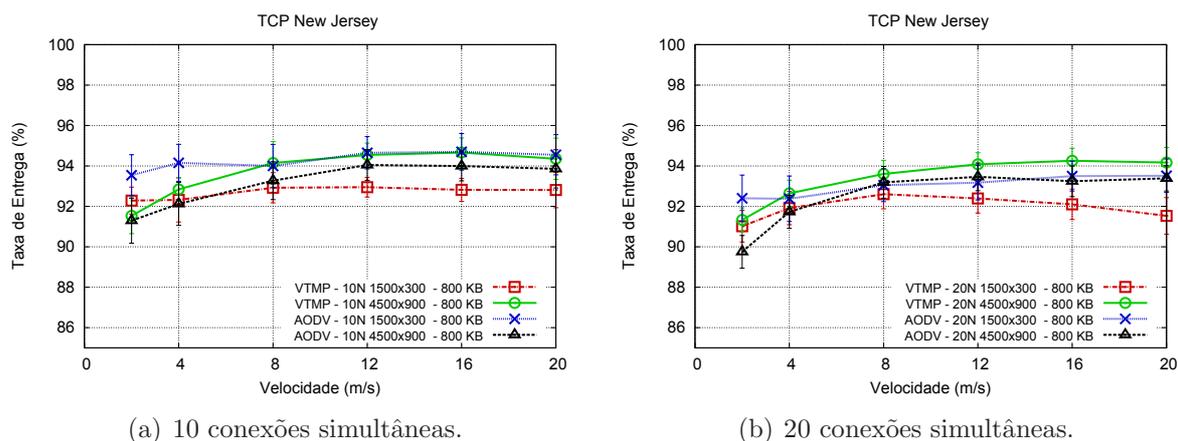


Figura 6.39: Taxa de Entrega considerando arquivos de 100 KB.

Na taxa de entrega o TCP NJ apresentou resultados diferentes dos resultados do TCP Reno.



(a) 10 conexões simultâneas.

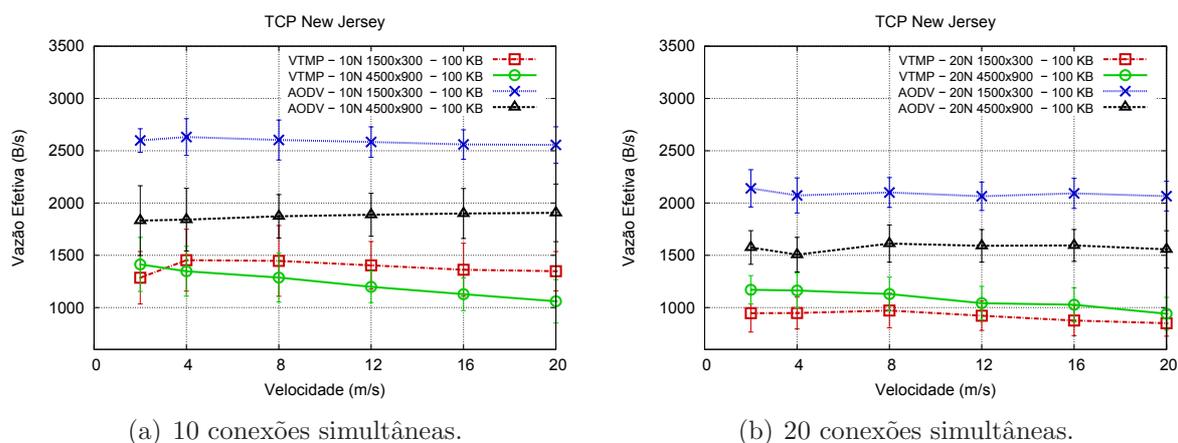
(b) 20 conexões simultâneas.

Figura 6.40: Taxa de Entrega considerando arquivos de 800 KB.

No cenário com transmissão de arquivos pequenos, de 100 KB, em uma rede densa, o uso do VTMP reduziu a taxa de entrega, mas em comparação com a taxa de entrega do TCP Reno essa redução foi menor. No cenário com a rede esparsa não há diferença significativa nos cenários com e sem o uso do VTMP. Isso ocorreu pois o TCP NJ calcula a largura de banda do caminho, possibilitando uma melhor tomada de decisão com relação a retransmissão de pacotes.

Na transmissão de arquivos maiores, de 800 KB, o uso do VTMP reduz em torno de 1% a taxa de entrega na rede densa. Na rede esparsa na média o uso do VTMP aumenta a taxa de entrega.

A vazão efetiva dos cenários com transmissão de arquivos de 100 e 800 KB é apresentada nas Figuras 6.41 e 6.42.

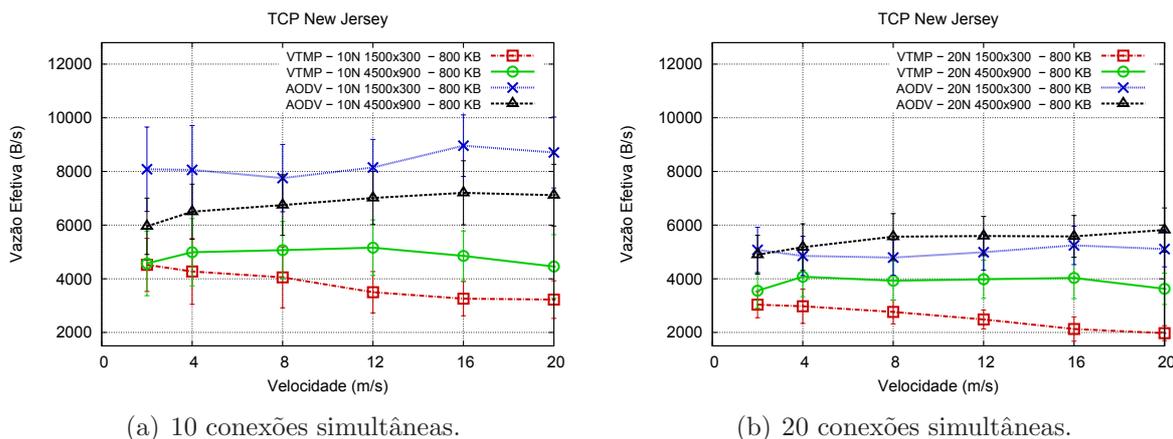


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.41: Vazão Efetiva considerando arquivos de 100 KB.

Na métrica vazão efetiva o uso do VTMP degradou o desempenho da rede, quando comparado



(a) 10 conexões simultâneas.

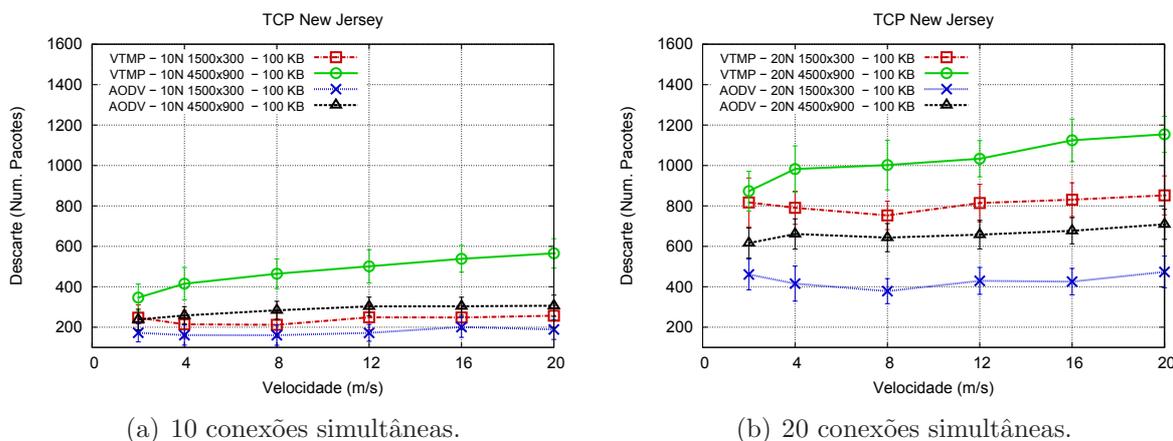
(b) 20 conexões simultâneas.

Figura 6.42: Vazão Efetiva considerando arquivos de 800 KB.

com os mesmos cenários sem o uso do VTMP.

No cenário com transmissão de arquivos de 100 KB o uso do VTMP reduz a vazão efetiva tanto na rede densa como na rede esparsa. Na média, no cenário com baixo tráfego de dados e uso do VTMP, Figura 6.41(a), a rede densa apresenta uma vazão efetiva maior que a rede esparsa. Ao aumentar o tráfego de dados, Figuras 6.41(b) e 6.42, a rede esparsa apresenta uma vazão efetiva maior. Isso ocorre pois o aumento no tráfego de dados na rede densa aumenta a disputa pelo acesso ao meio físico, ocasionando maior número de colisões e perdas de pacotes, o que na rede esparsa ocorre menos.

Nas Figuras 6.43 e 6.44 são apresentados os resultados da métrica descarte total de pacotes em cenários com transmissão de arquivos de 100 e 800 KB.

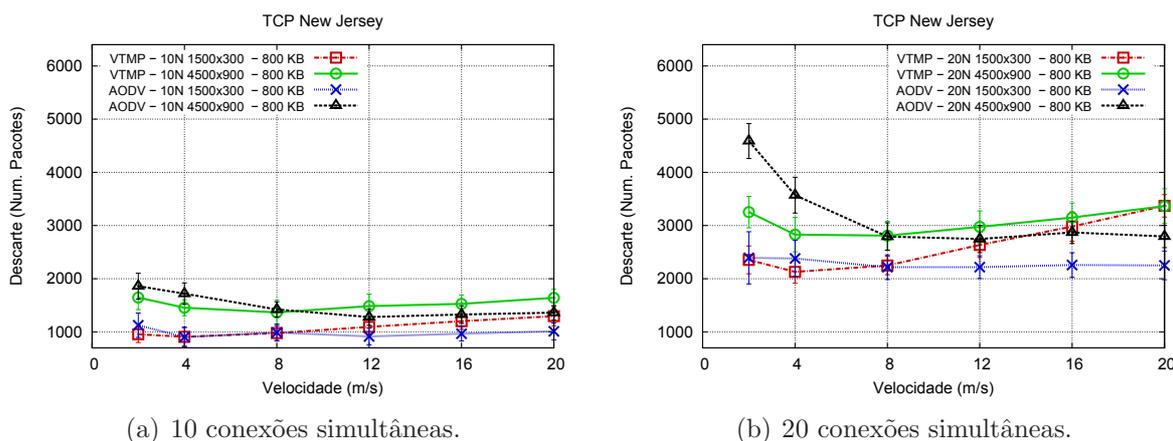


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.43: Descarte de pacotes considerando arquivos de 100 KB.

No geral o uso do do TCP NJ reduziu o descarte total de pacotes, em todos os cenários,



(a) 10 conexões simultâneas.

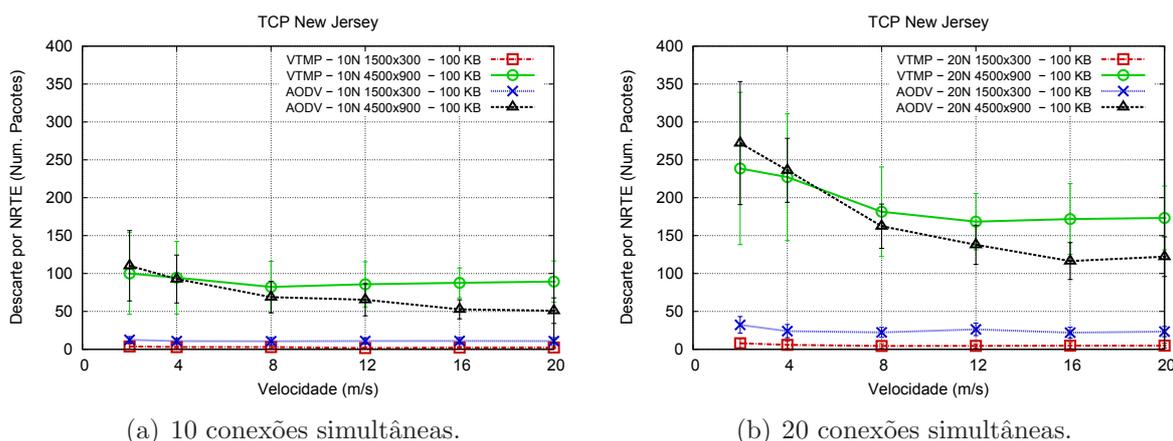
(b) 20 conexões simultâneas.

Figura 6.44: Descarte de pacotes considerando arquivos de 800 KB.

inclusive com o uso do VTMP. Nos cenários com transmissão de arquivos de 100 KB o uso do VTMP aumentou o número de descartes total de pacotes de dados. Esse aumento ocorreu por causa da sobrecarga gerada e pela baixa quantidade de dados enviados pela rede.

Com o aumento no número de pacotes de dados enviados, cenários com transmissão de arquivos de 800 KB, o descarte com o uso do VTMP é em alguns casos menor ou igual ao cenário sem o uso do VTMP, ao considerar o desvio padrão.

Nas Figuras 6.45 e 6.46 são apresentados os resultados da métrica de descarte de pacotes por falta de rota, em cenários com transmissão de arquivos de 100 e 800 KB respectivamente.

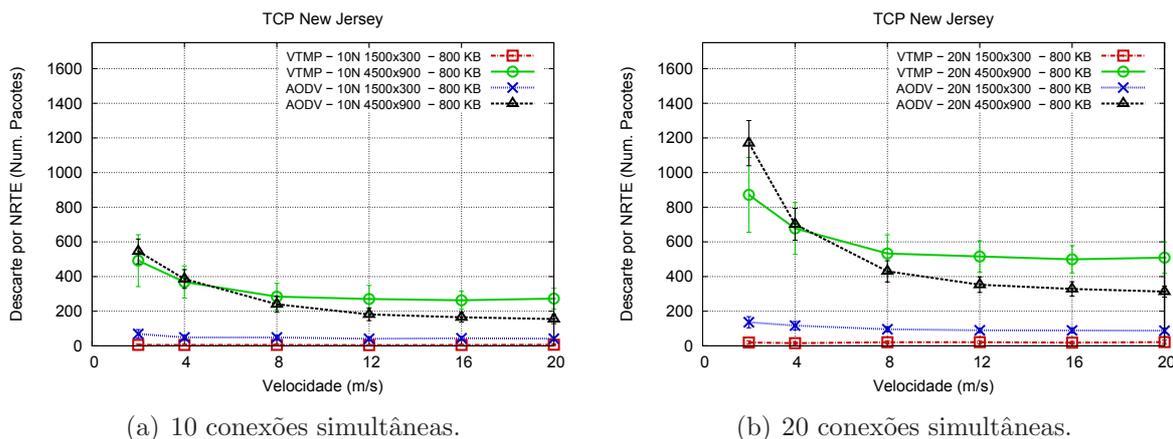


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.45: Descarte de pacotes por falta de rota considerando arquivos de 100 KB.

Os descartes por falta de rota na rede sem o uso do VTMP foram reduzidos com o uso do TCP NJ, em todos os cenários avaliados em comparação aos resultados obtidos pelo TCP Reno. Os cenários com uso do VTMP não tiveram diferença com o uso do TCP NJ.



(a) 10 conexões simultâneas.

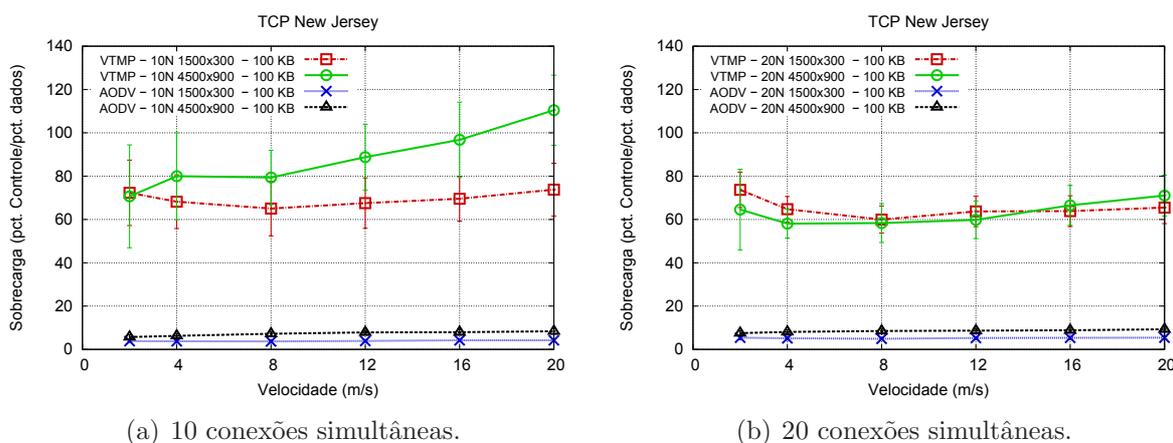
(b) 20 conexões simultâneas.

Figura 6.46: Descarte de pacotes por falta de rota considerando arquivos de 800 KB.

Em todos os cenários com alta densidade de nós o uso do VTMP reduziu o número de pacotes descartados por falta de rota. Nos cenários com a rede esparsa o uso do VTMP não apresentou diferença significativa ao cenário com baixa velocidade de movimentação dos nós. Quando a velocidade de movimentação máxima é superior a 8 m/s o uso do VTMP aumenta o número dos descartes por falta de rota.

O uso do TCP NJ sobre o VTMP não reduziu os descartes por falta de rota como ocorrido na rede sem o uso do VTMP. Isso ocorreu por conta da sobrecarga gerada pelo VTMP, que impede o funcionamento ótimo para o roteamento de mensagens.

Os resultados da métrica sobrecarga são apresentados nas Figuras 6.47 e 6.48, para os cenários de transmissão de arquivos de 100 e 800 KB respectivamente.

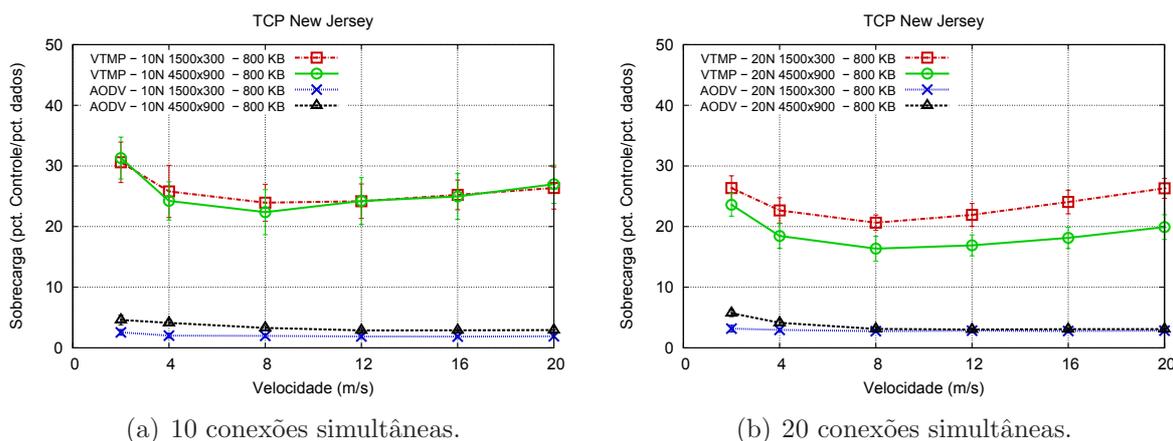


(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.47: Sobrecarga considerando arquivos de 100 KB.

O aumento da sobrecarga já era esperado na rede sem o uso do VTMP, pois independente



(a) 10 conexões simultâneas.

(b) 20 conexões simultâneas.

Figura 6.48: Sobrecarga considerando arquivos de 800 KB.

do envio dos dados a rede do VTMP mantém as rotas pró-ativamente. O uso da implementação NJ do protocolo TCP não afeta a sobrecarga da rede, quando comparada com as demais implementações do TCP.

Assim como nas demais avaliações a sobrecarga na rede com uso do VTMP só é alterada quando há um aumento na quantidade de dados transmitida durante a simulação.

Por meio dos resultados pode-se observar que o desempenho nos cenários sem o uso do VTMP obtiveram uma pequena melhora, quando comparado com os cenários com o uso do TCP Reno. Já os resultados nos cenários com o uso do VTMP essa diferença não é tão significativa.

Assim como nos resultados do TCP Reno, o TCP NJ apresentou melhores resultados em redes esparsas, em baixa velocidade de movimentação dos nós e alto tráfego de dados. Isso ocorre pois nestes cenários há menor quebra de rotas e o VTMP utiliza mais as rotas virtuais mantidas por ele.

## 6.5 Considerações Finais

Neste capítulo foram apresentados os resultados alcançados com o uso do VTMP em diferentes cenários. Por meio dos resultados pode-se observar que é possível a inserção do VTMP na camada de protocolos, desde que sejam aceitáveis a perda de desempenho com relação a algumas métricas, como o aumento do atraso, sobrecarga e redução da vazão efetiva.

O VTMP foi avaliado sob diferentes implementações do TCP. As implementações avaliadas foram o TCP Reno, o Tahoe, o Vegas e o New Jersey. No geral os resultados das diferentes implementações tiveram resultados similares, com exceção do TCP Vegas, que apresentou uma

redução no atraso e na taxa de entrega. Apesar do TCP NJ ser uma implementação do TCP para redes móveis seus resultados foram semelhantes aos resultados do TCP Reno.

Em suma o VTMP apresentou melhores resultados em redes esparsas, em baixa velocidade de movimentação dos nós e alto tráfego de dados.

## CAPÍTULO 7

### AVALIAÇÃO EXPERIMENTAL DO FUNCIONAMENTO DO VTMP SOB UMA APLICAÇÃO REAL

Neste Capítulo é apresentada a simulação de uma aplicação real sobre uma rede com o protocolo VTMP. O Objetivo deste teste não foi avaliar o desempenho, mas sim validar o funcionamento do VTMP.

Em [48] foi proposta a aplicação MADPastry, que é a implementação da *Distributed hash table* (DHT) para redes móveis. Uma DHT é um sistema de sobreposição de roteamento utilizado principalmente para o armazenamento de dados [10]. As DHTs convencionais são implementadas e executadas na camada de aplicação e são independentes das camadas inferiores, apenas utilizando os serviços dessas camadas para mandar e receber informações de outros nós pertencentes a DHT.

O trabalho desenvolvido em [48] sugere que: Para a DHT funcionar sobre uma rede MANET, é necessário que a DHT conheça a mobilidade dos nós, caso contrário ela não funcionará adequadamente. A solução proposta pelos autores foi implementar a DHT na camada de roteamento, alterando o funcionamento desta camada para o correto funcionamento da DHT. O MADPastry, alcançou resultados significativos, mas com o custo de modificar todo o funcionamento da DHT, adicionando custos para sua adoção e acabou por alterar totalmente sua implementação original.

Para validar o funcionamento de uma aplicação sobre o VTMP, foi utilizado uma rede de área 4500x900 metros, 128 nós, uso do protocolo VTMP, da DHT e do protocolo TCP Reno para envio de dados. Variou-se a velocidades de movimentação dos nós de 2 a 20m/s, sendo que o resultado obtido é a média de 35 simulações para cada velocidade de movimentação dos nós.

A simulação consiste em um cliente DHT solicitando um serviço para a DHT. A DHT propaga a solicitação em sua estrutura até o nó responsável pelo serviço solicitado. O nó, responsável pelo serviço, então estabelece uma conexão direta com o cliente que solicitou o serviço e envia um pacote de dados contendo o que foi solicitado.

As métricas apresentadas são o atraso e a taxa de entrega, para verificar a viabilidade de

execução da DHT. O Resultado da simulação é apresentado na Figura 7.1.

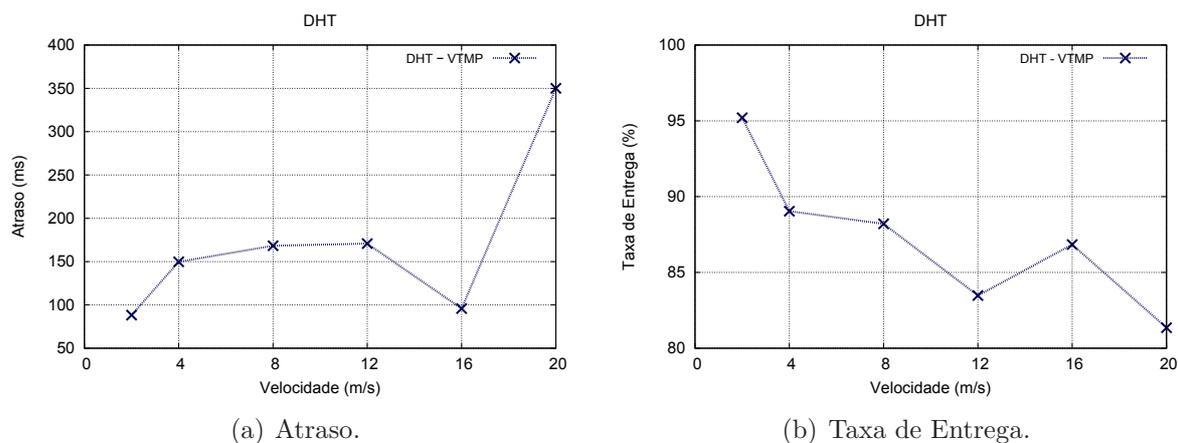


Figura 7.1: Avaliação da DHT.

Pode-se observar que o resultado do atraso (Figura 7.1(a)) ficou dentro do esperado, quanto comparado com os testes de envio de dados FTP sobre o VTMP. A taxa de entrega (Figura 7.1(b)) ficou abaixo do resultado alcançado no envio de dados FTP, um dos motivos pode ter sido a baixa quantidade de dados enviados o que pode influenciar negativamente no resultado. Com estes resultados fica validado a execução de um aplicação como a DHT sobre uma MANET com o uso do VTMP.

## CAPÍTULO 8

### CONCLUSÃO E TRABALHOS FUTUROS

As MANETs são redes sem fio compostas por dispositivos móveis heterogêneos, os quais não dependem de uma infra-estrutura pré-existente para funcionar, sendo mantidas por sistemas distribuídos e autônomos. Essas características atendem a necessidade crescente das pessoas de buscar a mobilidade aliada à conectividade, o uso das MANETs pode permitir a conexão à Internet em qualquer lugar sem a necessidade de uma infra-estrutura fixa para isso.

Para o uso das MANETs como uma extensão da Internet, espera-se que os protocolos e aplicações que são executadas na Internet funcionem corretamente nesta rede, sem grandes alterações. Como o protocolo TCP, um dos protocolos mais utilizados na Internet, apresenta problemas com a mobilidade dos nós, nesta dissertação foi proposto o VTMP, um protocolo para virtualizar de modo transparente uma rede fixa sobre a rede móvel.

O VTMP cria a rede virtual sobre a MANET escondendo a mobilidade da camada de transporte e aplicação, sem realizar alterações na camada de rede ou transporte. Ele apenas atua como protocolo de manutenção da rede fixa sobre o roteamento, encaminhando as mensagens por meio da rede virtual, até entregar a mensagem ao destino final. O VTMP foi implementado e avaliado no simulador NS-2, sendo que os protocolos de roteamento (AODV) e transporte (TCPs) utilizados na avaliação já estavam implementados no simulador.

Os cenários utilizados nas simulações, foram propostos para avaliar o custo de se manter uma rede virtual fixa em uma MANET, bem como validar o funcionamento do VTMP em situações que representassem melhor a realidade. Deste modo simulou-se cenários com a rede densa e outro com a rede esparsa, variando-se a velocidade de movimentação dos nós e carga de dados trafegando pela rede. O VTMP também foi avaliado sob diferentes implementações do TCP, para analisar a influência das melhorias realizadas no TCP e o impacto da escolha de uma dessas implementações para ser executada sobre o VTMP, visto que nossa proposta é de um protocolo independente da camada de transporte. As implementações do TCP utilizadas neste trabalho foram o TCP Reno, Tahoe, Vegas e New Jersey.

Com relação aos resultados o VTMP apresentou melhores resultados em redes esparsas, em

baixa velocidade de movimentação dos nós e alto tráfego de dados. No geral os resultados do VTMP sobre as diferentes implementações do TCP tiveram resultados similares, com exceção do TCP Vegas, que apresentou uma redução no atraso e na taxa de entrega. Apesar do TCP NJ ser uma implementação do TCP para redes móveis seus resultados foram semelhantes aos resultados do TCP Reno.

Por meio dos resultados alcançados nas simulações foi demonstrado que é possível a criação da rede fixa, desde que sejam aceitáveis seus custos. A vantagem do uso do VTMP é que ele pode ser utilizado sobre qualquer protocolo da camada inferior, sem ser necessário alterações. Uma desvantagem do uso do VTMP está no custo adicionado para o seu funcionamento, bem como a rigidez da estrutura do hipercubo, a qual pode limitar sua utilização.

Dentre as possíveis melhorias que podem ser implementadas no VTMP, que não foram realizadas neste trabalho, podemos citar: A adição de mecanismos no VTMP para monitoramento e retransmissão de pacotes descartados; Avaliação de novas estruturas virtuais para a rede fixa, a qual possa ser mais escalável; Implantação e validação sobre outros protocolos de roteamento; Avaliação mais detalhada do funcionamento de outras aplicações executadas sobre o VTMP.

## BIBLIOGRAFIA

- [1] A. Al-Dhelaan e B. Bose. Efficient fault tolerant broadcasting algorithm for the cube-connected cycles network. *Communications, Computers and Signal Processing, 1989. Conference Proceeding., IEEE Pacific Rim Conference on*, páginas 161 –164, june de 1989.
- [2] Luiz Carlos P. Albin, Antonio Caruso, Stefano Chessa, e Piero Maestrini. Reliable Routing in Wireless Ad Hoc Networks: The Virtual Routing Protocol. *Journal of Network and Systems Management*, 14(3):335–358, julho de 2006.
- [3] Stefano Basagni, Marco Conti, Silvia Giordano, e Ivan Stojmenovic, editors. *Mobile Ad Hoc Networking*. IEEE Press and John Wiley & Sons, Inc., Piscataway, NJ and New York, NY, April de 2004.
- [4] B. Bhargava. Peer-to-Peer File-Sharing Over Mobile Ad Hoc Networks. *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, páginas 104–108, 2004.
- [5] Ljubica Blažević, Silvia Giordano, e Jean-Yves Le Boudec. Self organized terminode routing simulation. *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, MSWIM '01*, páginas 81–88, New York, NY, USA, 2001. ACM.
- [6] Prosenjit Bose, Pat Morin, Ivan Stojmenović, e Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications, DIALM '99*, páginas 48–55, New York, NY, USA, 1999. ACM.
- [7] Lawrence S. Brakmo e Larry L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on selected Areas in communications*, 13:1465–1480, 1995.
- [8] Shan Chen, B. Bensaou, e Ka Lok Hung. Performance of different tcp variants in ieee 802.11 wlan and the tcp-wow algorithm. *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, páginas 1 –6, 30 2009-dec. 4 de 2009.

- [9] T. Clausen e P. Jacquet. Optimized Link State Routing Protocol (OLSR). *Internet RFC 3626*, 2003.
- [10] G. COULOURIS, J. DOLLIMORE, e TIM KINDBERG. *Sistemas Distribuídos: Conceitos e Projeto*. BOOKMAN COMPANHIA ED, 2007.
- [11] K. Day e A. Tripathi. A comparative study of topological properties of hypercubes and star graphs. *Parallel and Distributed Systems, IEEE Transactions on*, 5(1):31–38, jan de 1994.
- [12] Shuo Ding. A survey on integrating manets with the internet: Challenges and designs. *Computer Communications*, 31(14):3537–3551, 2008.
- [13] Sherif M. ElRakabawy, Alexander Klemm, e Christoph Lindemann. Tcp with gateway adaptive pacing for multihop wireless networks with internet connectivity. *Comput. Netw.*, 52(1):180–198, janeiro de 2008.
- [14] Kevin Fall e Sally Floyd. Simulation-Based Comparisons of Tahoe, Reno and SACK TCP. *SIGCOMM Comput. Commun. Rev.*, 26:5–21, July de 1996.
- [15] Hong fei Liu, Li jun Li, Zu yuan Yang, e Xi yue Huang. Tcp vegas\_m: A novel tcp algorithm in mobile ad hoc network. *ITS Telecommunications Proceedings, 2006 6th International Conference on*, páginas 629–633, june de 2006.
- [16] Zhenghua Fu, Xiaoqiao Meng, e Songwu Lu. How bad tcp can perform in mobile ad hoc networks. *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, páginas 298–303, 2002.
- [17] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, e Bobby V. Improving the performance of tcp over the atm-ubr service. *Computer Communications*, páginas 1042–1048, 1997.
- [18] Christian Grimm e Harald Schwier. Empirical Analysis of TCP Variants and Their Impact on GridFTP Port Requirements. *Proceedings of the Third International Conference on Networking and Services*, páginas 24, Washington, DC, USA, 2007. IEEE Computer Society.
- [19] Zygmunt J. Haas, Marc R. Pearlman, e Prince Samar. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. IETF Internet Draft, 1997.

- [20] Gavin Holland e Nitin Vaidya. Analysis of TCP performance over mobile ad hoc networks. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99*, páginas 219–230, 1999.
- [21] Yih-Chun Hu, Adrian Perrig, e David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2):21–38, janeiro de 2005.
- [22] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 18(4):314–329, agosto de 1988.
- [23] David B. Johnson e David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, páginas 153–181. Kluwer Academic Publishers, 1996.
- [24] Jon Postel. Transmission Control Protocol - DARPA Internet Program Protocol Specification. *Internet RFC 793*, 1981.
- [25] Md. Nazmul Islam Khan, Rashed Ahmed, e Md. Tariq Aziz. A survey of tcp reno, new reno and sack over mobile ad-hoc network. *International Journal of Distributed and Parallel Systems (IJDPS)*, 3, 2012.
- [26] Dzmitry Kliazovich e Fabrizio Granelli. Cross-layer congestion control in ad hoc wireless networks. *Ad Hoc Networks*, 4(6):687 – 708, 2006.
- [27] Young-Bae Ko e Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, julho de 2000.
- [28] M. Kovacevic. On torus topologies with random extra links. *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, volume 2, páginas 410–418 vol.2, 1996.
- [29] James F. Kurose e Keith w. Ross. *Redes de Computdores e a Internet: uma abordagem top-down*. Pearson Addison Wesley, São Paulo, SP, 3th edition, 2006.
- [30] V. Marbukh, K. Sayrafian-Pour, H. Mahboubi, A. Momeni, e A.G. Aghdam. Towards evolutionary-pricing framework for mobile sensor network self-organization. *Evolutionary Computation (CEC), 2010 IEEE Congress on*, páginas 1 –8, july de 2010.

- [31] Consolée Mbarushimana e Ali Shahrabi. E-TCP: Enhanced TCP for IEEE802.11e Mobile Ad Hoc Networks. *2009 15th International Conference on Parallel and Distributed Systems*, páginas 632–639, 2009.
- [32] Steven McCanne e Sally Floyd. NS Network Simulator, acessado em 2012. <http://www.isi.edu/nsnam/ns/>.
- [33] G.P. Millar, E.A. Panaousis, e C. Politis. Robust: Reliable overlay based utilisation of services and topology for emergency manets. *Future Network and Mobile Summit, 2010*, páginas 1 –8, june de 2010.
- [34] Seyed Mohsen Mirhosseini e Fatemeh Torgheh. Adhoctcp: Improving tcp performance in ad hoc networks, mobile ad-hoc networks. *Mobile Ad-Hoc Networks: Protocol Design*, páginas 121 – 138, jan de 2011.
- [35] Richard G. Ogier, Fred L. Templin, e Mark G. Lewis. Topology Broadcast Based on Reverse-Path Forwarding (TBRPF). *Internet RFC 3684*, 2004.
- [36] Charles E. Perkins. Ad Hoc Networking. capítulo Ad hoc networking: an introduction, páginas 1–28. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [37] Charles E. Perkins e Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24:234–244, October de 1994.
- [38] Charles Edward Perkins e Elizabeth Michelle Royer. Ad-hoc on-demand distance vector routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3):90–100, 1999.
- [39] Venugopalan Ramasubramanian, Zygmunt J. Haas, e Emin Gün Sirer. Sharp: a hybrid adaptive routing protocol for mobile ad hoc networks. *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '03*, páginas 303–314, New York, NY, USA, 2003. ACM.
- [40] Theodore S. Rappaport. *Comunicação Sem Fio: Princípios e Práticas*. Pearson Education do Brasil Ltda., São Paulo, SP, 2th edition, 2008.

- [41] J.P. Rohrer, A. Jabbar, E.K. Cetinkaya, E. Perrins, e J.P.G. Sterbenz. Highly-dynamic cross-layered aeronautical network architecture. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(4):2742–2765, october de 2011.
- [42] Y. Saad e M.H. Schultz. Topological properties of hypercubes. *Computers, IEEE Transactions on*, 37(7):867–872, jul de 1988.
- [43] I.K. Tabash, N. Ahmad, e S. Beg. A congestion window control mechanism based on fuzzy logic to improve tcp performance in manets. *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, páginas 21–26, oct. de 2011.
- [44] Andrew Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 4th edition, 2002.
- [45] K. Xu, Y. Tian, e N. Ansari. TCP-Jersey for Wireless IP Communications. *IEEE Journal on Selected Areas in Communications*, 22(4):747–756, maio de 2004.
- [46] Kai Xu, Ye Tian, e Nirwan Ansari. Improving TCP performance in integrated wireless communications networks. *Computer Networks*, 47(2):219–237, fevereiro de 2005.
- [47] Yuhua Yuan, Huimin Chen, e Min Jia. An optimized ad-hoc on-demand multipath distance vector(aomdv) routing protocol. *Communications, 2005 Asia-Pacific Conference on*, páginas 569–573, 2005.
- [48] Thomas Zahn e Jochen Schiller. MADPastry: A DHT Substrate for Practicably Sized MANETs. *Proc. of 5th Workshop on Applications and Services in Wireless Networks (ASWN2005)*, Paris, France, junho de 2005.