



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
CAMPUS LUIZ MENEGHEL

PAULO CESAR BARBIERI JUNIOR

**ESTUDO DA VIABILIDADE DAS MANETS COMO
EXTENSÃO DA INTERNET**

Bandeirantes
2014

PAULO CESAR BARBIERI JUNIOR

**ESTUDO DO USO DAS MANETS COMO EXTENSÃO
DA INTERNET**

Trabalho de Conclusão de Curso submetido a
Universidade Estadual do Norte do Paraná,
como requisito parcial para a obtenção do
grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Neimar Neitzel

Bandeirantes
2014

PAULO CESAR BARBIERI JUNIOR

**ESTUDO DO USO DAS MANETS COMO EXTENSÃO
DA INTERNET**

Trabalho de Conclusão de Curso
submetido a Universidade Estadual do
Norte do Paraná, como requisito parcial
para a obtenção do grau de Bacharel em
Sistemas de Informação.

COMISSÃO EXAMINADORA

Prof. Me. Neimar Neitzel
UENP – *Campus* Luiz Meneghel

Prof. Me. Luiz Fernando Legore do
Nascimento
UENP – *Campus* Luiz Meneghel

Prof. Me. Ricardo Gonçalves Coelho
UENP – *Campus* Luiz Meneghel

A todos aqueles que de alguma forma contribuíram para que a conclusão deste trabalho se tornasse possível.

AGRADECIMENTOS

Primeiramente a Deus que me ajudou nos momentos mais difíceis, nas madrugadas, nos momentos de angústia.

Aos meus amigos em especial, Victor, Guilherme, Fernando, Elielson, José Mário, Kelvin, Alexandre que me auxiliaram durante o curso, nos momentos tristes e felizes.

A banca avaliadora, em especial meu orientador Neimar Neitzel, pela dedicação, paciência e atenção em todos os momentos, sendo o principal responsável pelos créditos obtidos pelo projeto.

A minha família em especial, minha mãe Roseliana e meu pai Paulo Cesar, minha namorada Lívia que sempre estiveram do meu lado nos momentos mais difíceis de minha vida.

“O conhecimento e a informação são os recursos estratégicos para o desenvolvimento de qualquer país. Os portadores desses recursos são as pessoas.”

(Peter Drucker)

RESUMO

Com a grande taxa de usuários acessando a Internet por meio de dispositivos móveis é necessário a utilização de redes que possam trabalhar com topologia dinâmica, como uma *Mobile Ad Hoc Network* (MANET). Redes MANETS são redes sem fio e sem uma infraestrutura preexistente, onde os nós se conectam uns aos outros. Seus protocolos de roteamento distribuídos são responsáveis por gerenciada e manter as rotas de entrega de pacotes. Um dos protocolos mais utilizados na Internet é o protocolo de transporte TCP, que ao ser utilizado sobre redes sem fio sofre degradação de desempenho, devido às características de suas implementações e a taxa de perda de pacotes nessas redes. O problema do TCP é agravado quando aplicado em uma MANET pela alta mobilidade dos nós e conseqüentemente o descarte dos pacotes. Neste trabalho é apresentado um estudo do uso das MANETS como uma extensão da Internet, onde foram avaliados diferentes variantes do protocolo TCP em conjunto com os principais protocolos de roteamento.

Palavras-chave: TCP; MANET; Internet; *Network Simulator*; Protocolos de Roteamento.

ABSTRACT

With the high rate of users accessing the Internet through mobile devices using networks that can work with dynamic topology, as a Mobile Ad Hoc Network (MANET) is required. MANETs networks are wireless networks without a pre-existing infrastructure, where nodes connect to each other. Its distributed routing protocols are responsible for maintaining the routes managed and package delivery. One of the most used protocols on the Internet is the TCP transport protocol, which when used over wireless networks suffer performance degradation, due to the characteristics of their implementations and the loss rate of packets in these networks. The problem is compounded when the TCP implemented in a MANET by high mobility of nodes and consequently discarding packets. This study is the use of MANETs is presented as an extension of the Internet, which we assessed different variants of TCP in conjunction with major routing protocols.

Keywords: TCP; MANET; Internet; Network Simulator; Routing Protocols.

LISTA DE ABREVIATURAS E SIGLAS

ACK	<i>Acknowledgement.</i>
AODV	<i>Ad Hoc On-Demand Distance Vector.</i>
CWND	<i>Congestion Window.</i>
DARPA	<i>Defense Advanced Research Projects Agency.</i>
DSDV	<i>Destination-Sequence Distance-Vector.</i>
DSR	<i>Dynamic Source Routing.</i>
FK	<i>Full ACK.</i>
MANET	<i>Mobile Ad hoc Network.</i>
MPR	<i>Multipoint Relays.</i>
NAM	<i>Network Animator.</i>
NS	<i>Network Simulator.</i>
OLSR	<i>Optimized Link State Routing.</i>
PA	<i>Partial ACK.</i>
RREP	<i>Route Reply.</i>
RREQ	<i>Route Request.</i>
RTT	<i>Round Trip Time.</i>
RWND	<i>Receiver Window.</i>
TCP	<i>Transmission Control Protocol.</i>

LISTA DE ILUSTRAÇÕES

Figura 2.1: Exemplo de Rede MANET	20
Figura 2.2: Processo de Envio de Dados utilizando TCP	25
Figura 5.1: Atraso, considerando 1 conexão para o AODV.....	40
Figura 5.2: Atraso, considerando 5 conexões simultâneas para o AODV.....	40
Figura 5.3: Atraso, considerando 1 conexão para o DSR	41
Figura 5.4: Atraso, considerando 5 conexões simultâneas para o DSR	42
Figura 5.5: Atraso, considerando 1 conexão para o OLSR	42
Figura 5.6: Atraso, considerando 5 conexões simultâneas para o OLSR	43
Figura 5.7: Descarte de pacotes, considerando 1 conexão para o AODV	45
Figura 5.8: Descarte de pacotes, considerando 5 conexões simultâneas para o AODV	45
Figura 5.9: Descarte de pacotes, considerando 1 conexão para o DSR.....	46
Figura 5.10: Descarte de pacotes, considerando 5 conexões simultâneas para o DSR.....	47
Figura 5.11: Descarte de pacotes, considerando 1 conexão para o OLSR.....	48
Figura 5.12: Descarte de pacotes, considerando 5 conexões simultâneas para OLSR	49
Figura 5.13: Vazão efetiva, considerando 1 conexão para o AODV	50
Figura 5.14: Vazão efetiva, considerando 5 conexões simultâneas para o AODV....	50
Figura 5.15: Vazão efetiva, considerando 1 conexão para o DSR.....	51
Figura 5.16: Vazão efetiva, considerando 5 conexões simultâneas para o DSR	51
Figura 5.17: Vazão efetiva, considerando 1 conexão para o OLSR.....	52
Figura 5.18: Vazão efetiva, considerando 5 conexão para o OLSR.....	53
Figura 5.19: Taxa de entrega, considerando 1 conexão para o AODV	54
Figura 5.20: Taxa de entrega, considerando 5 conexões simultâneas para o AODV.	54
Figura 5.21: Taxa de entrega, considerando 1 conexão para o DSR.....	55
Figura 5.22: Taxa de entrega, considerando 5 conexões simultâneas para o DSR..	55
Figura 5.23: Taxa de entrega, considerando 1 conexão para o OLSR	56
Figura 5.24: Taxa de entrega, considerando 5 conexões simultâneas para o OLSR	57

Figura 5.25: Tempo total de envio de dados, considerando 1 conexão para o AODV	58
Figura 5.26: Tempo total de envio de dados, considerando 5 conexões simultâneas para o AODV	59
Figura 5.27: Tempo total de envio de dados, considerando 1 conexão para o DSR	60
Figura 5.28: Tempo total de envio de dados, considerando 5 conexões simultâneas para o DSR.....	60
Figura 5.29: Tempo total de envio de dados, considerando 1 conexão para o OLSR	61
Figura 5.30: Tempo total de envio de dados, considerando 5 conexões simultâneas para o OLSR	61

LISTA DE TABELAS

Tabela 3.1: Cenário de simulação utilizado por HENNA (2009).....	32
Tabela 3.2: Cenário de simulação utilizado por HOLLAND e VAIDYA (2002)	33
Tabela 3.3: Cenário de simulação utilizado por SEDDIK-GHALED et al. (2006)	34
Tabela 4.1: Quadro geral dos cenários de simulação	38
Tabela 5.1: Desempenho das implementações TCP sobre o protocolo de roteamento das MANETS.....	63

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Formulação e Escopo do Problema.....	16
1.2 Justificativa	17
1.3 Objetivo.....	17
1.3.1 Objetivos Específico	17
1.4 Metodologia	18
1.5 Organização do Trabalho.....	18
2 FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 <i>Mobile Ad Hoc Network</i>	20
2.2 Protocolos de Roteamento.....	21
2.2.1 <i>Dynamic Source Routing (DSR)</i>	21
2.2.2 <i>Ad Hoc On-Demand Distance Vector (AODV)</i>	23
2.2.3 <i>Optimized Link State Routing (OLSR)</i>	24
2.3 <i>Transmission Control Protocol (TCP)</i>	25
2.3.1 Início lento	27
2.3.2 Prevenção de Congestionamento	28
2.3.3 Retransmissão Rápida	28
2.3.4 Recuperação Rápida	28
2.3.5 TCP Reno.....	29
2.3.6 TCP Vegas	30
2.3.7 TCP New Jersey.....	30
2.4 Considerações Finais	31
3 TRABALHOS RELACIONADOS	32
3.1 <i>A Throughput Analysis of TCP Variants in Mobile Wireless Networks</i>	32
3.2 <i>Analysis of TCP Performance over Mobile Ad Hoc Networks</i>	33
3.3 <i>Effect of Ad Hoc Routing Protocols on TCP Performance within MANETS</i>	34
3.4 Considerações Finais	35
4 MATERIAIS E MÉTODOS.....	36
4.1 Materiais	36
4.2 Métricas de Avaliação	36
4.3 Parâmetros de Simulação.....	37
5 RESULTADOS E DISCUSSÕES	39
5.1 Atraso.....	39

5.2 Descarte de Pacotes.....	44
5.3 Vazão efetiva	49
Figura 5.15: Vazão efetiva, considerando 1 conexão para o DSR.....	51
5.4 Taxa de entrega.....	53
5.5 Tempo total de envio de dados.....	57
5.6 Considerações finais.....	62
6 CONCLUSÃO.....	64
6.1 Trabalhos Futuros.....	65
REFERÊNCIAS.....	66
APÊNDICE A.....	69

1 INTRODUÇÃO

No mundo atual, a Internet tomou proporções gigantescas, as vantagens proporcionadas pelo seu uso são inúmeras. Pode-se destacar como principais benefícios a busca pela informação de forma rápida, fácil e sua utilização para interligar computadores, que antes por estar situados em cidades ou até mesmo países diferentes não se comunicavam.

Seu uso foi intensificado com o advento da Web 2.0, que além da busca por informações, tornou possível a execução de aplicativos sem que o mesmo esteja devidamente instalado no dispositivo. Esse modo de utilização foi introduzido pela Computação em Nuvem (*Clouding Computer*), que segundo MACHADO (2009) é um modelo criado com o objetivo de fornecer serviços de fácil acesso e de baixo custo, tendo como garantia a disponibilidade e escalabilidade.

Outro fator que influenciou o acesso a Internet foi o maior poder de processamento que os dispositivos móveis atuais possuem e a possibilidade de realizar conexão sem fio com a Internet, que é encontrada facilmente em ambientes como restaurantes, shopping, bares e etc. Esse fator proporcionou o aumento de usuários conectados a Internet, fazendo com que ela alcance-se o patamar em que se encontra atualmente (POSSER, 2006).

A conexão de dispositivos móveis a Internet pode ser feita por dois tipos de rede, as que possuem infraestrutura fixa e as redes sem infraestrutura fixa (TANENBAUM, 2003).

Uma rede sem infraestrutura em que os dispositivos podem se mover livremente no ambiente é denominada Rede *Ad Hoc* Móvel (MANET). A conexão entre os dispositivos é realizada por meio da transmissão sem fio. Ao se utilizar rede MANET para conectar dispositivos, pode-se obter benefícios como a mobilidade dos nós, não ter necessidade de uma infraestrutura preexistente e fácil integração com outras tecnologias.

Essas redes possibilitam ainda maior conectividade dos dispositivos em ambientes que não se pode criar uma rede com infraestrutura física. Para gerenciar a rede os nós se interligam e fazem o papel de roteadores.

Um dos protocolos comumente utilizado na Internet é o TCP, que é de grande importância, pois a maioria das aplicações utiliza esse protocolo para a troca de

informações na Internet, o problema do uso do TCP é que o mesmo sofre grande perda de desempenho quando utilizado nas redes sem fio, o que se torna mais acentuado quando utilizado sobre uma MANET, devido à quebra de rotas causada pela alta mobilidade dos nós.

Dado este cenário de perda de desempenho do TCP sobre uma MANET, neste trabalho foi avaliado as diferentes implementações do protocolo TCP sobre diferentes protocolos de roteamento das MANETS. Com essa avaliação foi possível determinar qual implementação do TCP tem melhor desempenho sobre os protocolos de roteamento das MANETS.

1.1 Formulação e Escopo do Problema

O uso de redes sem fio para comunicação entre dispositivos degrada o desempenho do protocolo de Transporte TCP. Alguns destes problemas foram descritos por HOLLAND, VAIDYA (2002) e HENNA (2009) em seus trabalhos como: perdas de pacotes por erros de link, rotas para o destino inexistente ou congestionamentos.

Mesmo o protocolo TCP possuindo mecanismos que garantam a confiabilidade de entrega de pacotes, não é possível determinar o tipo de falha que está ocorrendo. Segundo HENNA (2009), os erros que ocorrem são considerados erros por congestionamentos e seu tratamento para esse problema é a redução da transmissão de dados impactando no desempenho geral do TCP.

Para melhorar o desempenho do TCP, diferentes implementações como New Reno, Reno, New Jersey, Vegas, foram criadas, visando classificar os descartes de pacotes e assim reagir do modo mais adequado.

Visto que existe diferentes implementações do TCP, que apresenta reações distintas ao funcionamento da rede, uma avaliação mais criteriosa se faz necessário. Deste modo, para a avaliação do funcionamento de uma conexão com a Internet sobre a MANET, foram realizadas simulações com diferentes implementações do TCP, visando definir qual implementação tem melhor desempenho sobre a MANET.

Outra questão que foi abordada são os protocolos de roteamento das redes MANET, existem diferentes protocolos de roteamento para MANET, mas nenhum deles é tido como padrão para essas redes. Deste modo, neste trabalho foi avaliado

o funcionamento das diferentes implementações do TCP sobre os protocolos de roteamento mais citados, os protocolos AODV, DSR e OLSR.

1.2 Justificativa

Rede Ad Hoc tornar-se uma grande ferramenta quando se necessita criar uma rede com o intuito de compartilhar Internet e seus recursos em um ambiente que não possui infraestrutura fixa.

Esse tipo de rede é comumente utilizada em cenários onde não se pode criar ou não se tem condições de criar uma infraestrutura para comunicação de dispositivos. BASAGNI (2004, p.411) destaca em seu livro, o uso das MANETs em operações especiais em apoio às atividades policiais, missões militares em um território hostil, encontros comerciais, tais como conferências, entre outros.

As MANETs se caracterizam por possuir topologia dinâmica onde os nós podem entrar ou sair da rede e também mover-se livremente. Essa característica influencia no desempenho do TCP pelo fato da alta mobilidade dos nós dentro da rede quebrar rotas do TCP, o que é identificado como congestionamento da rede pelo TCP.

HENNA (2009) em seu trabalho demonstra que dependendo da implementação do TCP e do protocolo de roteamento, é possível minimizar a perda de desempenho que o TCP sofre quando aplicado a uma rede sem fio.

Deste modo faz-se necessário uma avaliação mais criteriosa de diferentes cenários, é uma forma de reproduzir melhor a realidade, na qual o presente trabalho se sustenta para avaliar as principais implementações do TCP sobre uma MANET.

1.3 Objetivo

O objetivo deste trabalho foi a avaliação do uso da MANET como uma extensão da Internet, por meio da análise da melhor variante do protocolo TCP em conjunto com protocolos de roteamento.

1.3.1 Objetivos Específico

- Definição das métricas e cenários para avaliação;
- Simulação e avaliação dos diferentes cenários.

- Análise dos resultados, para definição do melhor TCP dadas as características da rede.

1.4 Metodologia

A natureza da pesquisa é exploratória a qual foi utilizada para determinar o uso das variantes do protocolo TCP, juntamente com os protocolos de roteamento das MANET como uma extensão da Internet, utilizando cenários que contenham uma abstração do mundo real. As técnicas utilizadas para o desenvolvimento do trabalho são pesquisas bibliográficas e simulações realizadas a partir de um simulador de redes.

Com relação a abordagem do problema este trabalho é desenvolvido seguindo uma abordagem quantitativa, uma vez que o foco está em determinar o melhor conjunto de protocolos para se trabalhar em determinado cenário.

Para alcançar os objetivos propostos nesta pesquisa foi necessário os seguintes passos metodológicos:

- a) Fundamentação Teórica: Abordagem de alguns temas como: funcionamento de uma rede do tipo MANET, estudo dos principais protocolos de roteamento para MANET, estudo das implementações do protocolo de transporte TCP e por fim definição das implementações do TCP que serão utilizadas neste trabalho.
- b) Estudo dos trabalhos relacionados para determinar quais protocolos de roteamento das MANET, implementações do protocolo TCP e métricas de avaliação serão utilizados nesse trabalho.
- c) Análise dos cenários encontrados na literatura para se obter um cenário o mais próximo possível da realidade para simulações.
- d) Realização das simulações dos cenários em um simulador de redes.
- e) Análise dos resultados obtidos com as simulações.

1.5 Organização do Trabalho

O trabalho apresenta a seguinte estrutura.

No Capítulo 2 são apresentados os estudos realizados sobre redes MANET incluindo seus protocolos de roteamento DSR, AODV e OLSR, também sobre o protocolo de transporte TCP começando a partir dos mecanismos de controle de

congestionamento Início Lento e Prevenção de Congestionamento, Retransmissão Rápida e Recuperação Rápida e por último as diferentes implementações do TCP utilizadas nas simulações. No Capítulo 3 estão descritos os trabalhos relacionados, utilizados como base para a escrita deste trabalho. No Capítulo 4 são apresentados os métodos e as métricas de avaliação e os parâmetros de configuração utilizados para gerar os resultados. No Capítulo 5 são apresentados os resultados através de gráficos e comentários dos respectivos resultados. O Capítulo 6 apresenta a conclusão obtida neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

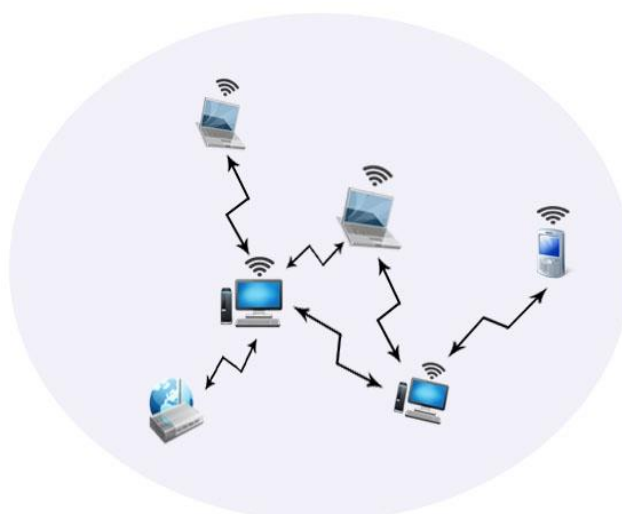
Neste capítulo são apresentados os conceitos sobre as MANETS que são pertinentes a este trabalho. Primeiramente é descrito as características de uma MANET, para então descrever os protocolos de roteamento utilizados neste trabalho, sendo que o funcionamento destes protocolos podem influenciar diretamente no desempenho dos protocolos de transporte. Por fim, são descritas as implementações dos protocolos de transporte TCP utilizadas na avaliação.

2.1 *Mobile Ad Hoc Network*

Segundo LIU (2011), uma rede *Mobile Ad Hoc Network* (MANET) é “uma rede na qual um grupo de dispositivos sem fios comunica-se sem o auxílio de uma infraestrutura de rede fixa.” Os dispositivos se conectam sem fio uns aos outros formando uma conexão ponto-a-ponto, outro fator importante é sua topologia que é dinâmica ocasionada pela alta movimentação e comunicação dos nodos. Nas MANETS não existe uma infraestrutura fixa todos os nodos possuem papel de roteador, encaminhando os pacotes recebidos de outros nodos, assim é possível fazer o compartilhamento de recursos na rede (TANEMBAUM, 2003, p.261).

Na Figura 2.1 é apresentado um exemplo de cenário onde os dispositivos estão conectados uns aos outros por meio de uma MANET.

Figura 2.1: Exemplo de Rede MANET.



Fonte: Adaptado de *Mobile ad hoc network* (BASAGNI, 2004, P.13).

Redes do tipo Ad Hoc podem ser aplicadas em cenários onde não há ou não se pode criar uma infra-estrutura fixa, cenários como: Ambientes Militares; Ambientes de Desastres; Redes Espontâneas; Extensão da Internet. Com a mesma facilidade de criação da rede ela também pode deixar de existir quando não mais necessária. (LIU, 2011).

2.2 Protocolos de Roteamento

O principal objetivo do protocolo de roteamento segundo BASAGNI (2004, p.21) “é o de estabelecimento de rotas corretas e eficientes entre um par de nós, de modo que as mensagens possam ser entregues de forma confiável e em tempo hábil”.

Os protocolos de roteamento podem ser classificados segundo suas características, neste trabalho foi utilizado protocolos de duas categorias: os protocolos pró-ativos e os protocolos reativos.

Protocolos pró-ativos mantêm tabelas de roteamento em cada um dos nós que fazem parte da rede, essas tabelas são responsáveis por guardar as rotas entre os nós. Protocolos pró-ativos mantêm rotas atualizadas para todos os nós, possibilitando o envio de dados sem necessitar da criação de rotas novas. Apesar das vantagens desses protocolos eles têm a desvantagem de gerar grande sobrecarga na rede, aumentando os gastos com energia dos nós.

Já os protocolos reativos ou sob demanda tem a característica de descobrir a rota quando necessário ao envio. Depois de realizado o processo de descoberta de rota, a mesma é mantida até que o destino torna-se inacessível, a rota não é mais utilizada ou a rota tenha se expirado.

Existem diferentes implementações dos protocolos reativos e pró-ativos. A seguir são descritas as principais implementações: O DSR (JOHNSON e MALTZ, 1996), AODV (PERKINS e ROYER, 1999) e OLSR (CLAUSEN e JACQUET, 2003).

2.2.1 *Dynamic Source Routing (DSR)*

FERNANDES (2003) descreve o protocolo DSR como sendo um “... protocolo de roteamento na fonte, que não utiliza mensagens de reconhecimento como as utilizadas no AODV, o que reduz a largura de banda necessária...”.

Para enviar um pacote ao nó destino, o nó remetente constrói e anexa ao cabeçalho do pacote a rota de envio, após isso é disparado o pacote ao primeiro nó da rota que está registrada. Quando um nó recebe um pacote, ele verifica se é o destino, caso não seja o destino, é realizada a retransmissão do pacote para o próximo nó identificado no pacote. (JOHNSON e MALTZ, 1996).

Os nós possuem uma memória chamada de *route cache* responsável por armazenar as rotas que foram descobertas e as rotas que passam pelo nó. Quando um pacote de dados precisa ser enviado, o nó primeiramente verifica em sua tabela se já existe uma rota estabelecida. Caso não exista uma o processo de descoberta de rotas é iniciado.

O mecanismo de descobertas de rotas inicia-se com a transmissão do pacote *Route Request* (RREQ) aos nós da rede por meio de *broadcast*, cada RREQ é identificado com o nó de origem, nó de destino, um identificador único que é identificado pela origem e uma lista de endereços que armazenará cada nó intermediário por onde o pacote passará.

Ao receber um RREQ ele precisa ser processado de acordo com alguns passos citados por FERNANDES (2003) em seu trabalho:

1. Ao receber o RREQ o nó verificar se é o destino, caso isso se concretize é enviado um pacote *Route Reply* (RREP) ao nó fonte que enviou o RREQ contendo uma cópia da rota localizada na lista de endereços do RREQ. Ao chegar o RREP o nó origem armazena as informações de rota em seu *route cache*.
2. Quando o nó receptor não é o destino, verificar se o RREQ já não foi processado ou se o seu endereço já está no *route cache* do nó. Caso alguma dessas condições sejam verdadeira o pacote é descartado.
3. Senão, procurar em seu *route cache* o caminho para o nó destino especificado no RREQ, ao encontrar é enviado um RREP para o nó destino com o caminho.
4. Caso o caminho necessário para entrega do pacote não esteja no *route cache* do nó, adiciona-se o próprio endereço do nó e retransmite o pacote por *broadcast*.

Ao detectar erro na rota o nó envia ao emissor do pacote de dados um pacote de erro contendo o endereço do nó que detectou o erro de rota e do nó que receberia o pacote. Esse pacote de erro deve retornar até o nó fonte, a rota de retorno pode ser construída a partir do *route cache* do nó.

Ao receber o pacote de erro de rota o nó fonte atualiza seu *cache route* removendo o salto com erro de todas as rotas que o utilizam.

2.2.2 Ad Hoc On-Demand Distance Vector (AODV)

No protocolo AODV os nós possuem tabelas de roteamento, na qual são armazenadas as descobertas. Cada linha da tabela possui informações para o próximo nó da rota e o nó destino, para que ao receber um pacote que não o pertence, verifique o destino e possa encaminhá-lo ao próximo nó da rota de destino. Quando o caminho para o destino não se encontra em sua tabela, o processo de descoberta de rotas é iniciado.

O início do processo de descoberta de rotas ocorre com o disparo do pacote *Route Request* (RREQ) que é identificado pelo endereço de sua origem e identificador de *broadcast*. (PERKINS e ROYER, 1999).

Quando um nó recebe um pacote RREQ ele precisa realizar alguns passos, descritos logo abaixo:

1. Realizar a verificação do endereço de origem e identificador de *broadcast* para identificar se o pacote em questão já não foi processado anteriormente, evitando que o pacote seja processado mais de uma vez.
2. Se o RREQ recebido não foi processado anteriormente, verifica se o nó atual é o destino do RREQ. Ao identificar que o nó atual é o destino do RREQ cria-se um pacote *Route Reply* (RREP), que é enviado para origem como resposta.
3. Caso o nó atual não seja o destino do RREQ, somente adicionar um registro na tabela de roteamento no qual o endereço de origem do pacote torna-se o endereço de destino, a fim de criar a rota reversa que o pacote RREQ percorreu. Após ser registrado o pacote é transmitido na rede novamente em *broadcast*.

4. Ao receber o RREQ o destino cria o pacote RREP o qual utiliza a rota reversa formada pelo RREQ para criar a rota da origem para o destino. Ao ser enviado o RREP passa por todos os nós que o RREQ passou antes de chegar até o destino, quando esses nós recebem o RREP registram em suas tabelas de roteamento qual é o próximo nó que é utilizado para se chegar ao destino, formando o caminho da origem para o destino.

Devido a alta mobilidade as MANETS sofrem de quebra de rotas, tornando necessário que um nó ao identificar a quebra de rota envie uma mensagem de erro para a origem do pacote.

Os nós enviam mensagens de *HELLO* aos seus respectivos vizinhos a fim de avisar que está ativo e que as rotas que estão contidas em sua tabela de roteamento ainda estão ativas. Esse processo somente é utilizado quando o nó fica muito tempo ocioso.

2.2.3 Optimized Link State Routing (OLSR)

O Protocolo OLSR é um Protocolo pró-ativo baseado em estado de enlace que periodicamente os nós inundam a rede com o estado de seus enlaces, isso ocorre, pois é preciso construir um mapa completo da topologia, (CLAUSEN e JACQUET, 2003).

O principal diferencial do protocolo OLSR para os outros protocolos pró-ativos é o uso de *Multipoint Relays* (MPR), quando um nó recebe um pacote de atualização de rotas e retransmite esse mesmo pacote para todos os seus vizinhos pelo método de inundação tradicional, acaba gerando uma grande sobrecarga na rede e fazendo com que cada nó da rede receba o mesmo pacote diversas vezes. Para minimizar esse problema o MPR delega a função de retransmissão de pacotes de controle apenas para determinados nós da rede. São os nós MPR que informam o estado dos links da rede. (CLAUSEN e JACQUET, 2003).

A métrica utilizada para escolha de um MPR segundo AREAL (2008) é a de “que o nó possa alcançar todos os seus vizinhos de dois saltos através do menor número de MPR possível.”

Ao entrar na rede o nó escolhe seu MPR, o nó MPR então envia mensagens contendo a informação do novo nó conectado a ele. Assim o novo nó conectado ao MPR passa a ser conhecido pela rede.

Nós que estão a mais de dois saltos da MPR necessitam manter informações sobre a topologia da rede, através de mensagens *Topology Control* (TC). Mensagens TC são geradas pelos nós MPR e são responsáveis por disseminar a lista de nós conectados aos MPR.

2.3 Transmission Control Protocol (TCP)

O TCP encontra-se alocado na camada de Transporte do Modelo TCP/IP, essa camada é responsável pelo controle de transporte de dados, TANENBAUM (2003, p. 49) descreve que esse protocolo foi projetado para trabalhar com diferentes topologias de redes, larguras de banda, retardos, tamanhos de pacote etc.

Para a utilização do TCP tanto o transmissor quanto o receptor devem estabelecer conexão ponto-a-ponto por um *sockets*, o qual contém o ip do host e o número da porta.

Outro fator importante é o modo pelo qual os dados são transmitidos em uma conexão TCP, utiliza-se o modo *Full-duplex* de envio no qual permite o tráfego de informações em ambas as direções ao mesmo tempo.

Figura 2.2: Processo de Envio de Dados utilizando TCP



Fonte: Adaptado de Buffers TCP de envio e de recepção (KUROSE, 2010, p. 175).

A Figura 2.2 descreve como é realizado o envio de dados considerando dois processos que vamos chamar de **A** e **B**, depois de criada a conexão entre **A** e **B**, **A** envia através do socket uma cadeia de dados. A partir desse momento o TCP se torna responsável pelo envio dos dados, esses dados são armazenados em seu *buffer* de envio. Os dados são então retirados do *buffer*, alocados em segmentos e enviados para o destino.

O tamanho dos segmentos segundo KUROSE (2010, p. 175) é definido pela unidade máxima de transmissão, utilizando como base o tamanho do maior quadro de camada de enlace que pode ser enviado pelo hospedeiro remetente.

Cada porção de dados do cliente é combinada a um cabeçalho TCP chamados de segmentos TCP, que encapsula dentro de datagramas IP e o envia ao destino. O destino ao receber os segmentos, armazena-o no *buffer* de recepção e a aplicação lê a cadeia de dados esse *buffer*.

O congestionamento em uma rede se dá quando, segundo CAVALCANTI (2005), a memória dos roteadores fica cheia, fazendo com que não seja possível armazenar os pacotes que chegam para serem processados.

O TCP possui controle de congestionamento fim-a-fim, isso indica que a presença de congestionamento é descoberta pelos sistemas finais com base em observações de comportamento da rede. A identificação do congestionamento é realizada de acordo com mecanismos chamados de estouros de temporizadores e reconhecimento duplicado. (TANEMBAUM, 2003, p. 298).

Estouro de temporizador ocorre sempre que o reconhecimento de um pacote é enviado e não chega ao receptor em um tempo pré-determinado (*timeout*). O funcionamento desse mecanismo é iniciado quando o pacote de dados é enviado, ao ser enviado um temporizador é acionado, ao receber o pacote o receptor deverá enviar a confirmação de recebimento do pacote *acknowledgments* (ACK), esse ACK deverá retornar ao nó remetente antes do *timeout*. Ao acontecer o *timeout* duas suposições podem ser feitas, a primeira é que o pacote não chegou ao destino e a segunda é que o ACK se perdeu durante a transmissão, ao perceber que o ACK não chegou o nó remetente envia o pacote novamente.

Para se calcular o tempo de *timeout* o TCP faz uso de uma estimativa chamada *round trip time* (RTT) que é baseado no tempo de ida de um pacote e no tempo de volta de um ACK, o RTT sofre mudanças de acordo com o tempo, pois há mudanças de rotas e do tráfico e para gerenciar isso, o TCP tem a responsabilidade de monitorar tais mudanças e modificar sempre que necessário o tempo de *timeout* para que não haja estouro dos temporizadores indevidamente.

Reconhecimento duplicado ocorre quando três ACK's são recebidos para um mesmo segmento, isso indica que o segmento anterior ao segmento que está sendo reconhecido foi perdido e o receptor informa isso mandando confirmações para o último pacote recebido, ou seja, o receptor sabe que antes do pacote que está entregue tem outro pacote anterior a este que ainda não foi entregue então, o

destinatário faz requisição para o reenvio desse pacote que está faltando. (TANEMBAUM, 2003, p. 234).

CAVALCANTI (2005) descreve que além da janela de recepção, *Receiver Window* (RWND), indicando a quantidade de segmentos que o receptor pode receber sem confirmação, o remetente mantém uma janela de congestionamento, *Congestion Window* (CWND), utilizada para impor uma limitação adicional à quantidade de dados que um nó pode enviar em uma conexão.

O papel da janela de congestionamento é o de controlar o fluxo sobre a rede, já a janela de recepção é controlar todo o fluxo que é imposto pelo receptor, existem quatro algoritmos responsáveis pelo controle de congestionamento: *Slow Start*, *Congestion Avoidance*, *Fast Retransmit* e *Fast Recovery*. (STEVENS, 1997). A seguir esses algoritmos serão detalhados.

2.3.1 Início lento

O algoritmo de Início Lento (*Slow Start*) é utilizado pelo TCP emissor para fazer o controle dos dados que são distribuídos pela rede, seu mecanismo de funcionamento se baseia de acordo com a seguinte observação, a taxa em que novos pacotes de dados são adicionados na rede deve ser a mesma em que as confirmações são enviadas pelo receptor. (STEVENS, 1997).

No início o algoritmo Início Lento possui CWND pequena que vai aumentando exponencialmente, isso ocorre da seguinte forma, ao enviar um segmento pela rede, é necessário que o ACK do mesmo retorne ao remetente, se o ACK voltar antes do *timeout* é enviado mais dois segmentos e novamente é aguardado a ACK dos mesmos, ao retornarem antes do *timeout* enviam-se mais quatro segmentos pela rede que assim sucessivamente vai crescendo até que:

1. Aconteça a perda de segmento, apontada por *timeout*.
2. CWND seja igual a RWND.
3. CWND é menor que *threshold*.

Ao ser reconhecido um congestionamento por meio de *timeout* (recepção de ACK's duplicados), o valor *threshold* é alterado para um valor que reflete a metade da janela de congestionamento atual. Ao ser alterado o valor da janela de

congestionamento é reiniciada, e com o tempo o emissor vai aumentando novamente a janela de congestionamento exponencialmente. (STEVENS, 1997).

2.3.2 Prevenção de Congestionamento

O algoritmo de Prevenção de Congestionamento (*Congestion Avoidance*) incrementa a CWND de um seguimento a cada RTT, fazendo com que o crescimento da CWND seja linear, esse processo continua sendo executado até que os reconhecimentos continuem chegando antes do fim de seus temporizadores. (CAVALCANTI, 2005).

Esse algoritmo segundo STEVENS (1997) tem a responsabilidade de prevenir o congestionamento, que pode ocorrer quando vários fluxos de dados chegam a um determinado roteador cuja sua capacidade seja excedida, sobrecarregando-o e causando perda de segmentos.

2.3.3 Retransmissão Rápida

O algoritmo de Retransmissão Rápida é iniciado quando o nó receptor recebe três ACKs duplicados. STEVENS (1997) descreve que um ou dois ACKs duplicados podem ser segmentos fora de ordem, mas quando três ou mais ACKs duplicados chegam é um forte indício de descarte de pacote.

Ao receber três ACKs o algoritmo de Retransmissão Rápida retransmite o segmento solicitado, não esperando pela expiração do RTT.

2.3.4 Recuperação Rápida

O algoritmo de Recuperação Rápida (*Fast Recovery*) entra em ação logo depois da execução do algoritmo de Retransmissão Rápida, com o papel de reduzir o congestionamento da rede evitando a necessidade do acionamento da técnica de Início Lento (STEVENS, 1997). Ao executar a Retransmissão Rápida, a Recuperação Rápida executa o algoritmo de Prevenção de Congestionamento, esse processo é utilizado segundo CAVALCANTI (2005) pelo motivo de que ao receber ACK's duplicados pode-se também deduzir que ainda existem dados trafegando entre dois nós, sendo que é possível o receptor informar que ainda existem segmentos intermediários não recebidos e pela qual a única maneira que o receptor tem para receber novos segmentos é a de informar essa informação.

A janela de congestionamento durante a execução da Recuperação Rápida é aumentada com a recepção de uma confirmação duplicada para que seja possível, o receptor poder continuar recebendo pacotes que tenham sido perdidos e possa corrigir suas falhas.

A partir da definição do TCP diferentes implementações como Reno, Vegas, New Jersey, foram criadas no intuito de combater a perda de desempenho que o TCP sofre quando aplicado em redes sem fio.

Nas subseções seguintes são descritas as implementações do TCP utilizadas nesse trabalho.

2.3.5 TCP Reno

O TCP Reno surgiu como uma melhora do TCP Tahoe (FALL e FLOYD, 1996), que foi uma das primeiras versões modificadas do TCP que surgiu. HENNA (2009) descreve que o TCP Reno é uma implementação utilizada pela grande maioria das redes atuais.

Seu funcionamento baseia-se no uso de diferentes algoritmos de controle de congestionamento como Prevenção de Congestionamento, Recuperação Rápida, Retransmissão Rápida e Início Lento, algoritmos esses que já foram descritos anteriormente. Para estimar a largura de banda disponível em rede é realizada a exploração de perda de pacotes.

Ao constituir uma conexão TCP o algoritmo de Início Lento é iniciado, ao iniciar esse processo o tamanho da CWND aumenta exponencialmente até que seja detectado o congestionamento, ao ser detectado o algoritmo de Prevenção de Congestionamento inicia os mecanismos Retransmissão Rápida e Recuperação Rápida, para o TCP Reno um congestionamento ocorre quando é feita a recepção de três ACK's duplicados ou quando ocorre *timeout*.

Ao utilizar esses dois mecanismos, há melhora de desempenho do TCP Reno ao utilizar *timeout* como indicação de congestionamento da rede. (HENNA, 2009).

FALL e FLOYD (1996) demonstram que o desempenho do TCP Reno é afetado quando há perdas consecutivas de pacotes sobre a rede, pelo fato do nó remetente reduzir a CWND até a confirmação de todos os pacotes perdidos, fazendo com que ocorra a redução da taxa de transmissão.

Outro problema destacado por FALL e FLOYD (1996) é a perda de pacotes ao se utilizar o algoritmo de Recuperação Rápida em casos de congestionamento, pois esse algoritmo necessita de um tempo maior para identificar o congestionamento.

2.3.6 TCP Vegas

O TCP Vegas é o que possui a estimativa de banda mais eficiente, comparados a outras versões do TCP. LAWRENCE et al. (1995) menciona que o TCP Vegas é uma modificação de implementação realizada sobre o TCP Reno, essa modificação foi realizada no modo como o mecanismo de prevenção de congestionamento trabalha. (HENNA, 2009).

O TCP Reno trata o congestionamento de modo reativo, enquanto que o TCP Vegas reage pró - ativamente tentando identificar o congestionamento para que seja possível suavizar seus efeitos. (LAWRENCE et al, 1995).

O TCP Vegas utiliza os algoritmos, Início Lento e Retransmissão Rápida do TCP Reno, e é no algoritmo de Início Lento que o TCP Vegas realiza mudanças no processo, quando se inicia o envio de dados, o algoritmo não reconhece o estado em que a rede está não sendo possível identificar se está livre de congestionamento.

O mecanismo de Prevenção de Congestionamento utilizado pelo TCP Vegas é diferente do mecanismo utilizado pelo TCP Reno. O TCP Vegas diminui sua CWND após iniciada a detecção de congestionamento, a fim de evitar perdas de pacotes, não diminuindo sua CWND desnecessariamente. Já o TCP Reno inicia o mecanismo de Prevenção de Congestionamento e aumenta sua CWND até que seja detectada alguma perda de pacotes na rede. (HENNA, 2009).

2.3.7 TCP New Jersey

O TCP New Jersey foi criado para ser utilizado em redes sem fio com uma abordagem pró-ativa ao congestionamento. Segundo HENNA (2009), o TCP New Jersey tem a função de distinguir perdas de pacotes por congestionamento e reagir de forma adequada, para isso utiliza dois mecanismos, Estimativa de Largura de Banda e Aviso de Congestionamento que são responsáveis por ajudar o remetente a identificar a causa da perda de pacotes no roteador intermediário.

As alterações realizadas no TCP New Jersey causaram melhorias significativas no processo de estimativa da largura de banda e congestionamento, sendo possível calcular o tamanho da janela de congestionamento de forma dinâmica, variando de acordo com a estimativa de largura de banda. A responsabilidade pelo cálculo da largura de banda é do nó receptor, para descartar qualquer influência que pode ocorrer devido a atrasos na entrega de pacotes dados e perda de pacotes ACK. (XU et al, 2004).

Através da estimativa gerada para largura de banda o transmissor pode aumentar ou diminuir sua janela de congestionamento. Para avisar sobre o congestionamento o TCP New Jersey possibilita que ao identificar um congestionamento os roteadores informem através de um bit encontrado nos pacotes enviado por ele indicando que a fila do roteador está acima do ideal.

2.4 Considerações Finais

Neste capítulo foram apresentados os principais conceitos sobre MANET e como os protocolos de roteamento trabalham para estabelecer rotas de entrega, também encontra-se descrito o protocolo TCP com seus mecanismos de controle de congestionamento e algumas de suas principais implementações.

Ao ser utilizado em redes sem fio o TCP sofre de grande perda de desempenho por tratar todas as falhas na rede como congestionamento, diminuindo sua janela de transmissão, o que se agrava quando aplicado a uma rede MANET que possui topologia altamente dinâmica.

O principal objetivo das implementações do TCP demonstradas é identificar as falhas que ocorrem, antes de executar o processo de Prevenção de Congestionamento onde, o tamanho da janela de transmissão é reduzido e conseqüentemente o desempenho do TCP é afetado.

3 TRABALHOS RELACIONADOS

Neste capítulo é apresentada uma revisão dos trabalhos que avaliam o desempenho do TCP sobre MANET. Na literatura há diversos trabalhos descritos que realizam comparações sobre diferentes implementações do protocolo TCP e protocolos de roteamento das MANETs.

3.1 A *Throughput Analysis of TCP Variants in Mobile Wireless Networks*

O trabalho de HENNA (2009) analisa o desempenho das variantes do TCP sobre os protocolos de roteamento das MANETS, a fim de verificar qual variante do TCP sobre o protocolo de roteamento possui o melhor desempenho com a grande perda de pacotes ocasionada pela alta mobilidade dos nós.

Tabela 3.1: Cenário de simulação utilizado por HENNA (2009).

Parâmetros	Valores
Número de nós	50
Área	1000x1000m
Velocidade dos nós	2, 5, 10, 15 e 20m/s
Tempo de pausa	0s.
Modelo de mobilidade	<i>Randon waypoint</i>
Protocolo MAC	802.11
Protocolo de roteamento	AODV, DSR e DSDV
Tamanho da fila do roteamento	50
Protocolo de Transporte	TCP Vegas e New Jersey
Quantidade de dados transmitida	2Mbps
Conexões	25
Taxa Aleatória de Perda de Pacotes (%)	0.5,1,5,10,20,30,40,50

Utilizando os parâmetros de simulação definidos na Tabela 3.1, HENNA (2009) analisou por meio da métrica de vazão efetiva, o impacto que a mobilidade dos nós causa no desempenho do TCP Vegas e New Jersey.

Através desta análise pode-se concluir que todas as combinações realizadas com o TCP Vegas e New Jersey sobre os protocolos de roteamento AODV, DSR e DSDV das MANETS, sofrem de grande perda de desempenho quando a mobilidade

aumenta, mas observou-se também que o TCP Vegas tem melhor desempenho que o TCP New Jersey quando combinado ao cenário definido.

3.2 Analysis of TCP Performance over Mobile Ad Hoc Networks

HOLLAND e VAIDYA (2002) em seu trabalho analisam o impacto que a quebra de rotas causa no desempenho do TCP. Para melhorar o desempenho do TCP, foi utilizada a técnica de notificação explícita e avaliada através da métrica de vazão efetiva. Todos os resultados gerados são baseados no cenário de simulação definido na Tabela 3.2, que utilizam o TCP Reno sobre o protocolo de roteamento DSR, segundo os autores esses protocolos são amplamente utilizados e estudados, sendo assim prováveis candidatos para implementação em redes Ad Hoc comerciais.

Tabela 3.2: Cenário de simulação utilizado por HOLLAND e VAIDYA (2002).

Parâmetros	Valores
Número de nós	30
Área	1500x300m
Velocidade dos nós	2, 10, 20 e 30m/s
Tempo de pausa	0s.
Modelo de mobilidade	<i>Randon waypoint</i>
Protocolo MAC	802.11
Protocolo de roteamento	DSR
Tamanho do pacote TCP	1460 bytes
Protocolo de Transporte	TCP Reno
Quantidade de dados transmitida	2Mbps
Conexões	1

Os autores concluíram que o desempenho do TCP cai significativamente quando a quebra de rotas é provocada pela movimentação dos nós. Isso ocorre devido à incapacidade do TCP reconhecer a diferença entre a quebra de rota e o congestionamento da rede. Também foi observado que ao se utilizar a notificação de falhas de ligação explícita (ELFN), pode-se melhorar o desempenho do TCP. O ELFN proporciona ao remetente TCP informações sobre rota e falhas de ligação, evitando que falhas por quebra de rota sejam interpretadas como congestionamento.

3.3 Effect of Ad Hoc Routing Protocols on TCP Performance within MANETS

No trabalho de SEDDIK-GHALED et al. (2006) é realizado um estudo de diferentes tipos de protocolos de roteamento que possuem características diferentes como os proativos/reativos, baseados em vetor de distância/estado de link e roteamento de origem. Os autores analisaram os efeitos que os protocolos de roteamento têm sobre o desempenho do TCP dentro das MANETs. O estudo levou em consideração a relação de dois fatores, que segundo os autores são comuns e podem afetar o desempenho do TCP quando aplicado as MANETs, são eles, a escolha do protocolo de roteamento e a mobilidade dos nós.

Tabela 3.3: Cenário de simulação utilizado por SEDDIK-GHALED et al. (2006).

Parâmetros	Valores
Número de nós	20
Área	670x670m
Velocidade dos nós	5, 15 e 30m/s
Modelo de mobilidade	<i>Random waypoint</i>
Protocolo MAC	802.11
Protocolo de roteamento	DSDV, OLSR, AODV e DSR.
Protocolo de Transporte	Reno, New Reno, Sack, WestwoodNR e Vegas.
Quantidade de dados transmitida	2Mbps
Conexões	14

Os resultados foram gerados através do cenário definido na Tabela 3.3 e avaliados pela média do consumo de energia e vazão da rede.

Através das simulações realizadas os autores concluíram que, o TCP é influenciado pela natureza dinâmica das MANETs, isso ocorre devido a alta mobilidade dos nós nesse tipo de rede. Também observaram que nem sempre a alta mobilidade dos nós é um fator que resulta em degradação do desempenho do TCP, podendo ser às vezes um auxílio para os protocolos de roteamento das MANETS, pois restabelece links quebrados de forma mais rápida. Por fim, classificaram o DSDV como tendo o melhor desempenho em todas as variantes do TCP dentro de uma MANET.

3.4 Considerações Finais

Foi apresentada neste capítulo uma revisão da literatura, de alguns trabalhos que analisam o desempenho do TCP sobre as MANETs.

A maioria dos trabalhos analisa o desempenho utilizando a métrica de vazão efetiva da rede para as implementações do TCP e de protocolos de roteamento das MANETS, com o objetivo de determinar a implementação do TCP que possui o melhor desempenho com a alta mobilidade dos nós quando aplicado ao cenário definido.

Neste trabalho foi realizada uma análise mais criteriosa das implementações do TCP sobre os protocolos de roteamento das MANETs, considerando além da vazão efetiva, outros fatores que influenciam o desempenho do TCP como: descarte de pacotes de dados, tempo de envio de dados e taxa de entrega. Para cada um destes fatores é utilizada uma métrica de avaliação, melhor descrita no Capítulo 4.

4 MATERIAIS E MÉTODOS

Neste capítulo é descrito os materiais e métricas de avaliação utilizados para criar e compilar os dados gerados pelas simulações, assim como os parâmetros de configuração que foram definidos para gerar as simulações.

4.1 Materiais

O simulador utilizado foi o Network Simulator (NS) na versão 2.34, o NS é um simulador de eventos discretos que tem por objetivo auxiliar na análise do comportamento das redes, criando-a em um ambiente virtual. Oferece ainda suporte para simulações utilizando protocolos TCP, de roteamento e multicast em redes com e sem fio. (MCCANNE e FLOYD, 2014).

Atualmente o NS está em sua versão 3, mas para o presente projeto a utilização do NS 2.34 satisfaz todas as necessidades exigidas, possuindo as versões mais estáveis das diferentes implementações do TCP que foram utilizadas no projeto. Para as redes MANETS somente o protocolo OLSR necessitou ser adicionado após a instalação do NS 2.34 através de um patch¹.

A versão do NS 2.34 foi utilizada na plataforma Linux, através do sistema operacional Ubuntu 13.04². Este sistema operacional foi escolhido, pois possibilita que alterações nas configurações do NS sejam realizadas de forma rápida e simples do que comparado a mesma versão do NS distribuída para a plataforma Windows. Outro fator importante para a escolha da plataforma *Linux*, foi a utilização do *Shell Script* em conjunto com a linguagem de interpretação *AWK* (AHO et. 1987) na criação do processo de rotinas de simulação em conjunto com o NS, sendo possível assim, automatizar o processo de criação da simulação e compilação dos dados.

4.2 Métricas de Avaliação

Para avaliar o desempenho do protocolo TCP sobre os protocolos de roteamento foram selecionadas as seguintes métricas de avaliação:

¹ O patch pode ser encontrado no site do Instituto de Informática da UFRGS (<http://www.inf.ufrgs.br/~wlccordeiro/resources/olsr/>).

² Para mais informações acesse: <http://releases.ubuntu.com/13.04/>.

- Taxa de Entrega: é a porcentagem de pacotes de dados entregues corretamente no destino, comparando o total de pacotes de dados enviados com o total de pacotes recebidos corretamente.
- Vazão Efetiva: É dada pelo total de dados enviados entre dois nós, considerando apenas pacotes de dados e desconsiderando pacotes de manutenção de rede. O cálculo é realizado através da soma dos *bytes* que um nó enviou dividido pelo tempo de envio dos dados, o resultado é apresentado em *kiloBytes*/segundo. Já nessa métrica quanto maior for a vazão dos dados, mais rápida a informação será transferida de um nó para o outro.
- Atraso: Definido pelo cálculo do tempo gasto para um pacote de dados ser entregue, o cálculo basicamente é feito com o valor de tempo que o pacote foi recebido pelo destino, subtraindo o valor do tempo inicial em que o mesmo pacote foi enviado pelo emissor, sendo representado por milésimos de segundos (ms). Através do atraso é possível determinar o quanto tempo os pacotes levam para ir da origem até o destino.
- Tempo Total de Envio de Dados: Demonstra o tempo total em segundos (s), que a rede utilizou para enviar todos os pacotes de dados que foram enviados pelos nós.
- Descarte Total de Pacotes: Refere-se aos pacotes que foram descartados durante a simulação, independente do motivo pelo qual o pacote foi descartado. A avaliação do número de pacotes descartados na rede é importante, pois quanto maior o número de pacotes descartados maior é o gasto com retransmissão de pacotes e pior é o desempenho do TCP.

4.3 Parâmetros de Simulação

Para melhor avaliar o resultado dos protocolos sobre as MANETs, em todas simulações foi considerado um ambiente móvel com uma área total de 1000x1000 contendo 50 nós, a velocidade de movimentação dos nós foi variada de 2 m/s até 12 m/s, para melhor avaliar diferentes níveis de mobilidade dos nós, a movimentação dos nós foi definido através do método *Random Waypoint*, que gera padrões de movimentação aleatórios considerando a velocidade máxima definida. O tempo de

pausa (tempo que o nó fica parado em uma posição após se movimentar) foi definido como 10 s.

A Tabela 4.1 mostra os parâmetros gerais utilizados nas simulações.

Tabela 4.1: Quadro geral dos cenários de simulação.

Parâmetros	Valores
Número de nós	50
Área	1000x1000m
Raio de alcance da transmissão	250m ²
Velocidade dos nós	2, 4, 8 e 12m/s
Tempo de pausa	10s
Modelo de mobilidade	<i>Random waypoint</i>
Protocolo MAC	802.11
Protocolo de roteamento	AODV, DSR e OLSR
Tamanho da fila do roteamento	50
Largura de banda	11 Mb
Protocolo de transporte	TCP Reno, Vegas e New Jersey
Janela de congestionamento do TCP	Padrão
Quantidade de dados transmitida	500KB e 2MB
Tamanho do pacote	576 e 1500 bytes

Nas simulações variou-se a quantidade de conexões de dados estabelecidas entre 1 e 5, com o intuito de avaliar o impacto do aumento de número de usuários ativos na rede. Todos os nós se conectam a um nó único com isso, pode-se simular como é ter somente um ponto único de acesso à Internet.

O tamanho de pacote foi definido em 576 e 1500 bytes, para avaliar o impacto da fragmentação dos dados a serem enviados. O tamanho total do arquivo a ser transmitidos foi definido em 2mb para simular o acesso à páginas web com diferentes tamanhos. Para cada um dos diferentes cenários, foram realizadas 35 simulações, em cada uma das 35 simulações o posicionamento e movimentação dos nós é aleatória.

5 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentadas as avaliações do impacto das diferentes implementações do TCP sobre os protocolos de roteamento das MANETS, utilizando os parâmetros de configuração definidos na Tabela 4.1 considerando apenas a carga de dados de 2 MB, os resultados das simulações com transmissão de 500 KB podem ser analisados no Apêndice A.

Para demonstrar os resultados gerados através das métricas descritas no item 4.2, foi utilizado o gráfico de linha. Cada gráfico contém três cenários simulados, sendo cada linha do gráfico a representação de um cenário. As marcações verticais de cada ponto no gráfico representam o desvio padrão da métrica avaliada, obtidos através de 35 simulações realizadas para cada ponto.

Os cenários encontrados nos gráficos é o resultado obtido através da métrica avaliada com o uso do TCP New Jersey, Reno e Vegas sobre os protocolos de roteamento o AODV, DSR e OLSR, variando a quantidade de conexões em 1 e 5 para o tamanho do pacote TCP de 576 e 1500 bytes.

Para cada métrica avaliada foi criado um item, contendo os gráficos dispostos lado a lado para uma melhor visualização do impacto causado pela modificação do tamanho do pacote TCP sobre os protocolos de roteamento.

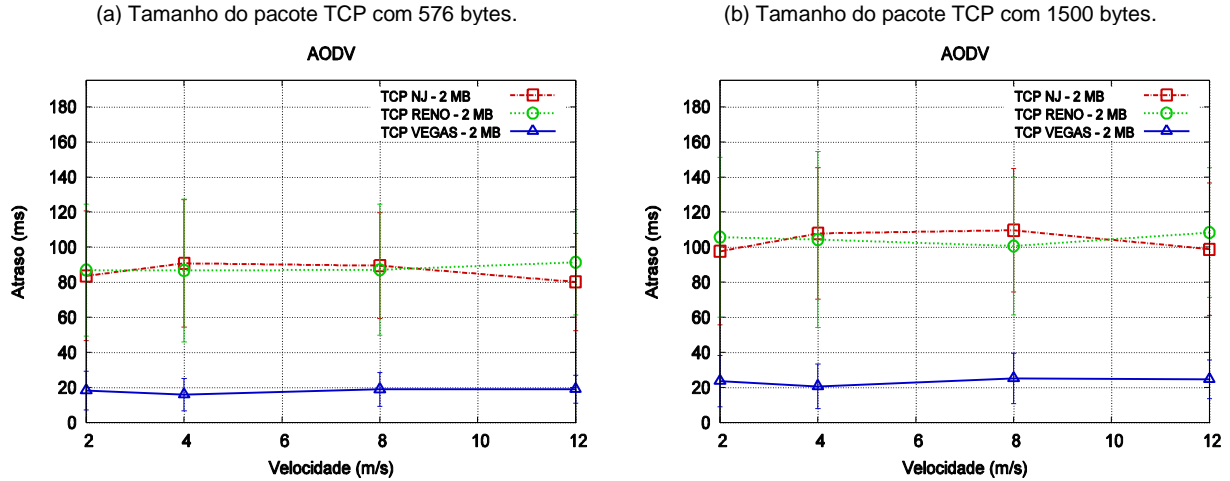
5.1 Atraso

Nesta seção é apresentada a avaliação dos protocolos de roteamento sobre as diferentes implementações do TCP utilizadas por este trabalho, através da métrica de atraso.

Na Figura 5.1 é apresentado o atraso na entrega de pacotes de dados, em um cenário com o envio de arquivos de 2 MB para 1 conexão, utilizando o AODV como protocolo de roteamento para as diferentes implementações do TCP. No gráfico pode-se observar que o TCP Vegas sobre o AODV tem o melhor desempenho apresentando o menor atraso, já o TCP New Jersey e Reno não têm diferença significativa. Para o tamanho do pacote TCP em 576 bytes (Figura 5.1(a)), utilizando 1 conexão o aumento médio no atraso do TCP Vegas foi de 18ms com os nós movendo-se a 2 m/s, já com os nós movendo-se a 12 m/s não houve diferença significativa. Com o tamanho do TCP definido em 1500 bytes (Figura 5.1(b)) os

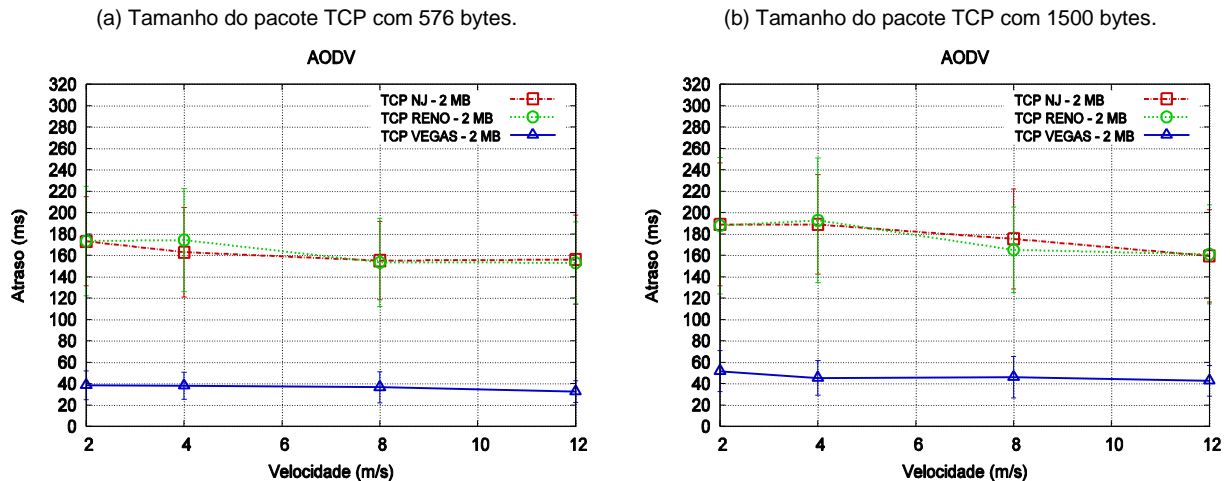
resultados são muito similares ao levar em consideração o desvio padrão de cada resultado.

Figura 5.1: Atraso, considerando 1 conexão para o AODV.



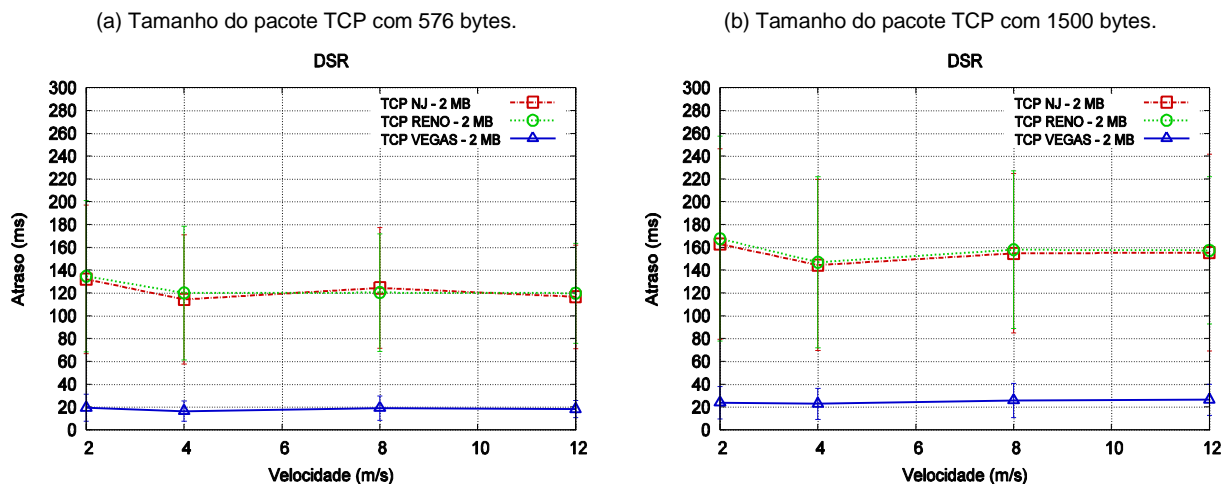
Com o aumento do número de conexões para 5, utilizando um tamanho de pacote TCP em 576 bytes, o atraso do TCP Vegas sobe para uma média de 40ms, a diferença entre o TCP New Jersey e Reno para o Vegas na média é de 120ms com os nós movendo-se a 12 m/s, como apresenta a Figura 5.2(a). Na Figura 5.2(b) é possível observar com o tamanho do pacote em 1500 bytes que o TCP Vegas têm aumento de 10 ms de atraso, quando comparado aos pacotes de 576 bytes. Ao analisar o desempenho do TCP New Jersey e Reno na Figura 5.2, pode-se visualizar que seu desempenho se mantém na média de 160 ms de atraso com os nós movendo-se a 12 m/s.

Figura 5.2: Atraso, considerando 5 conexões simultâneas para o AODV.



A Figura 5.3 apresenta o atraso na entrega de pacotes de dados para o protocolo de roteamento DSR, utilizando 1 conexão. Podemos observar que o TCP Vegas sobre o DSR possui desempenho maior com os nós movendo-se a 2m/s o TCP Vegas tem um atraso de 20 ms e a 12 m/s o resultado é semelhante, já o TCP New Jersey e Reno possuem desempenho equivalente, também é possível observar que com a maior mobilidade dos nós o desempenho do TCP New Jersey e Reno tem uma melhora, como se pode observar na Figura 5.3(a). Para o tamanho do pacote em 1500 bytes, o TCP Vegas tem um atraso médio de 23 ms com os nós movendo-se a 2 m/s e 26ms para 12 m/s, tendo aumento no atraso comparado ao TCP definido em 576 bytes, observando a figura 5.3(b) também é possível visualizar que o desempenho do TCP New Jersey e Reno continua semelhante, porém com aumento no atraso, devido à definição de 1500 bytes para o tamanho do pacote.

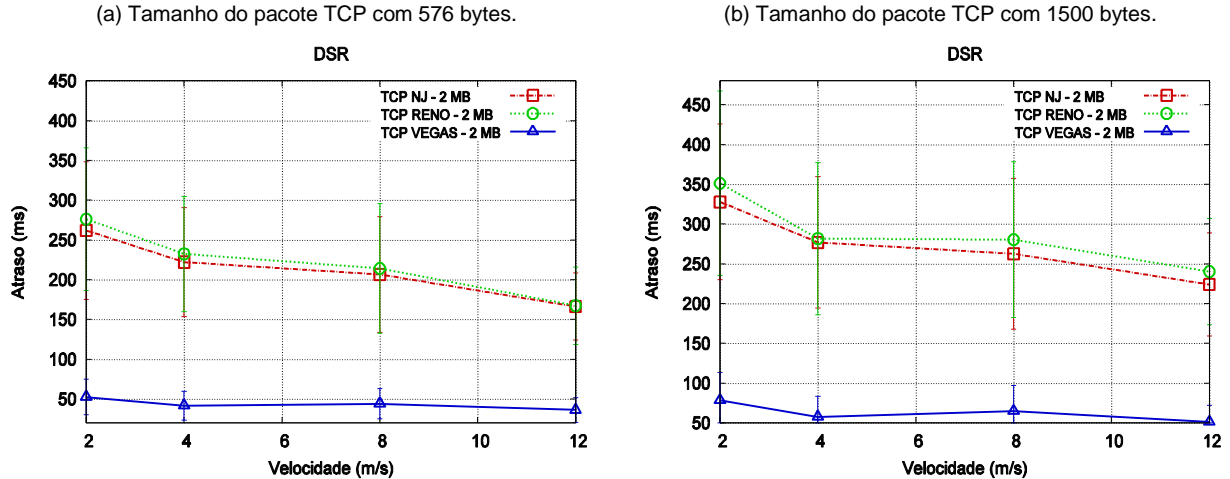
Figura 5.3: Atraso, considerando 1 conexão para o DSR.



Ao utilizar 5 conexões o TCP Vegas tem um aumento no atraso, tendo uma melhora quando os nós movem-se a 12 m/s. O desempenho do TCP New Jersey e Reno possui diferença a 2 m/s, sendo que a 12m/s o atraso médio é semelhante a 113 ms. Ao definir em 1500 bytes o tamanho do pacote (Figura 5.4(b)) é possível visualizar que há aumento no atraso de todas as implementações do TCP. Sendo que as piores médias ocorrem quando os nós movem-se a 2 m/s, o TCP Reno possui a pior média de atraso (350ms) e o TCP Vegas a melhor (78 ms), quando a movimentação alcança 12 m/s a média de atraso decresce para 58 ms do TCP Vegas e 240 ms para o TCP Reno, sendo assim é possível determinar na Figura 5.4 que quanto maior a movimentação dos nós menor é o atraso na entrega de pacotes ao se utilizar o DSR como protocolo de roteamento e que com o tamanho do pacote

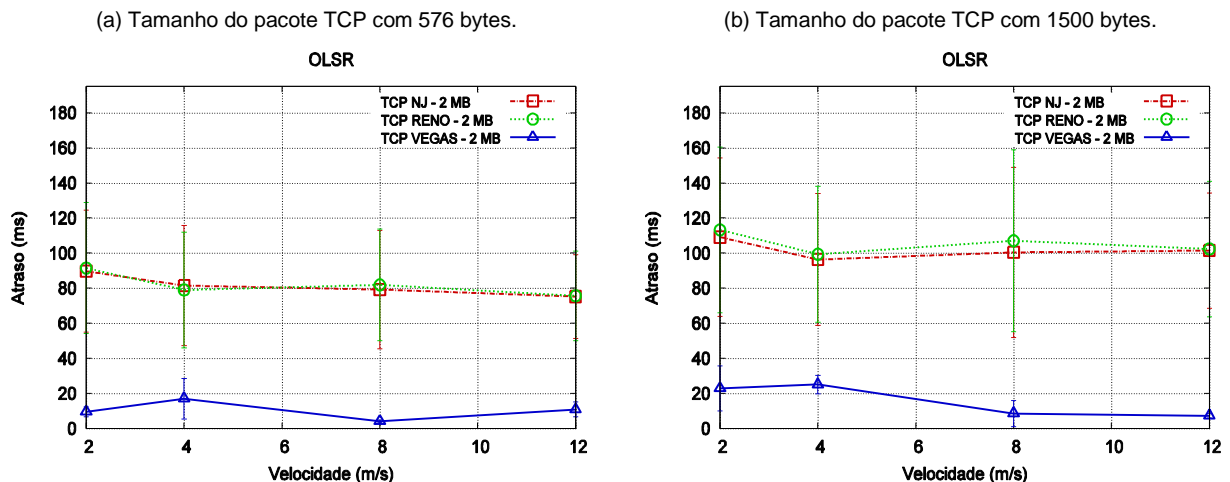
TCP definido em 576 bytes os atrasos são menores do que comparados ao TCP definido em 1500 bytes.

Figura 5.4: Atraso, considerando 5 conexões simultâneas para o DSR.



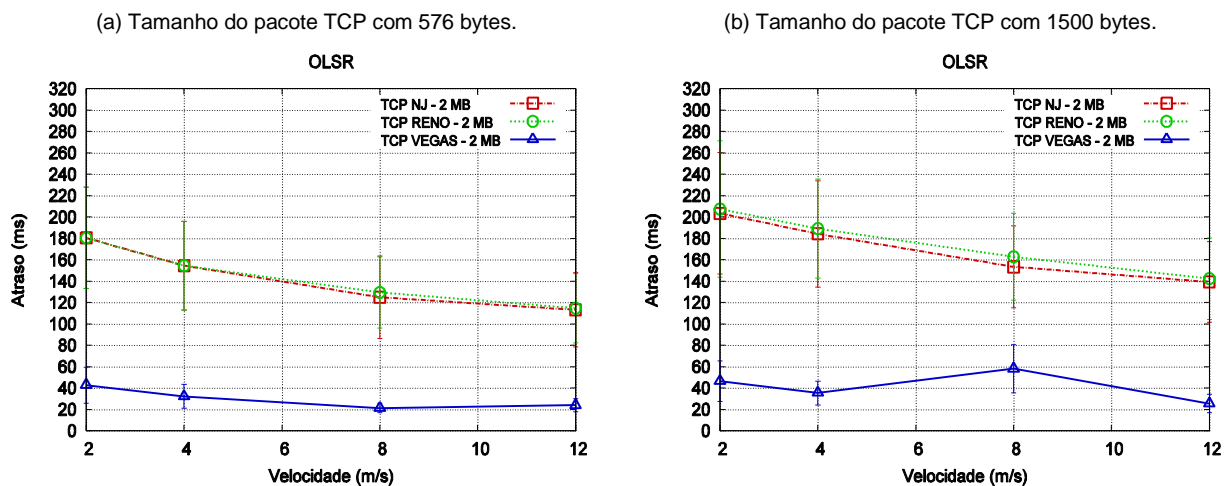
Na Figura 5.5 é apresentada a média de atraso ao utilizar o OLSR como protocolo de roteamento para as implementações do TCP, com o tamanho do pacote definido em 576 (Figura 5.5(a)) o TCP Vegas é que obtém o melhor desempenho, seu atraso com os nós movendo-se a 2 m/s é de 10ms, sofrendo variações de desempenho em velocidade maiores que 2m/s. O desempenho do TCP New Jersey e Reno é semelhante, possuindo melhora com a maior movimentação dos nós. Com o tamanho do pacote definido em 1500 bytes (Figura 5.5(b)), observa-se que há aumento na média de atrasos, comparado ao tamanho do pacote de 576 bytes. Mesmo com o aumento na média, o TCP Vegas é que possui o melhor desempenho, já o TCP New Jersey e Reno possuem desempenho semelhante.

Figura 5.5: Atraso, considerando 1 conexão para o OLSR.



Ao definir 5 conexões e o aumento da mobilidade dos nós no OLSR, é possível observar na Figura 5.6 que a média de atraso tende a diminuir em todas as implementações do TCP. A implementação que obteve o melhor desempenho foi o TCP Vegas com o tamanho do pacote TCP definido em 576 bytes (Figura 5.6(a)), possuindo uma média de 42 ms com os nós movendo-se a 2 m/s e com movimentação de 12 m/s o TCP Vegas melhorou sua média em torno de 19ms, já o TCP New Jersey e Reno possuem melhora de 66ms a 12 m/s. Com a definição do tamanho do pacote TCP em 1500 bytes (Figura 5.6(b)), houve aumento na média de atrasos de todas as implementações do TCP comparado a Figura 3.6(a), o TCP Vegas com movimentação de 2 m/s possui média de 46 ms, com o aumento da movimentação para 12 m/s o TCP Vegas melhora seu desempenho em 25 ms.

Figura 5.6: Atraso, considerando 5 conexões simultâneas para o OLSR.



No geral, os melhores resultados obtidos na métrica de atraso na entrega de pacotes de dados, foram obtidos através do uso do OLSR como protocolo de roteamento das MANETs utilizando o TCP Vegas como protocolo de transporte. A média de atraso obtida pelo TCP Vegas utilizando o OLSR como protocolo de roteamento foi de 10 ms para o tamanho do pacote TCP definido em 576 e de 20 ms para o tamanho de 1500 bytes.

Outro fator observado foi que para os protocolos de roteamento DSR e OLSR, quanto maior for o número de nós transmitindo e recebendo na rede e maior velocidade de movimentação dos mesmos, menor é o atraso na entrega de pacotes de dados, já com o aumento do tamanho do pacote TCP de 576 bytes para 1500 bytes, ocorre o aumento da média de atrasos, esses fatores ficam mais visível no TCP New Jersey e Reno.

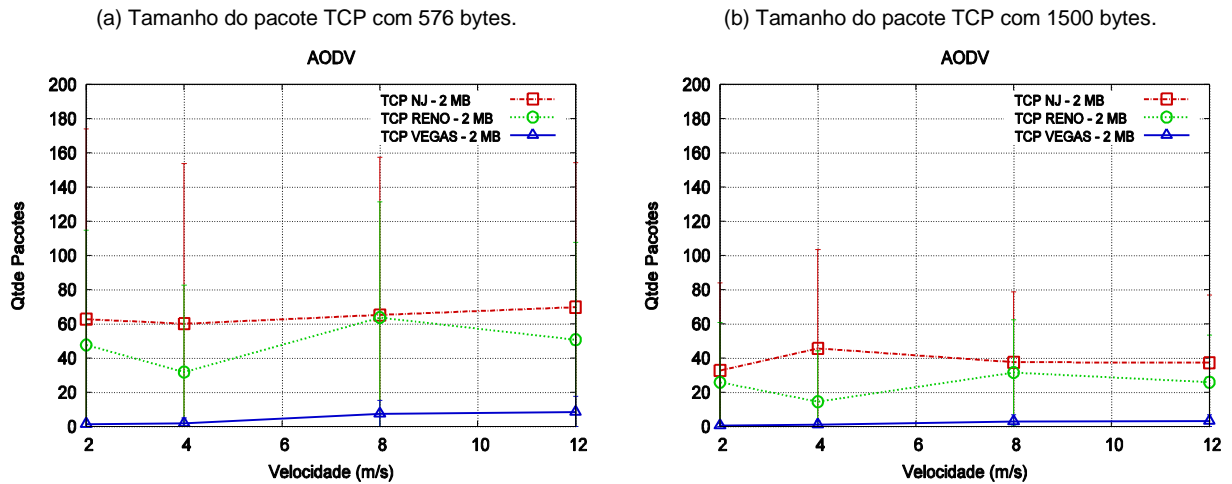
5.2 Descarte de Pacotes

Nesta seção é apresentada a avaliação dos protocolos de roteamento sobre as diferentes implementações do TCP utilizadas por este trabalho, através da métrica de descarte de pacotes.

A Figura 5.7 apresenta a avaliação das implementações TCP sobre o protocolo de roteamento das MANETS o AODV, em um cenário com somente 1 conexão. Também é possível observar que praticamente não houve descarte pelo TCP Vegas e que o TCP New Jersey é que possui o pior desempenho. Com o tamanho do pacote TCP definido em 576 bytes (Figura 5.7(a)), o TCP Vegas tem uma média de descarte de somente 1 pacote com os nós movendo-se a 2 m/s e 8 pacotes a 12 m/s. O TCP que obteve o pior desempenho foi o TCP New Jersey com média de descarte de 62 pacotes em 2 m/s e 69 pacotes a 12 m/s, já o TCP Reno possui média de 50 pacotes descartados com a movimentação dos nós em 2 m/s e desempenho semelhante a 12 m/s, também é possível visualizar que o TCP Reno tem melhor desempenho quando os nós estão se movimentando a 4 m/s, melhorando sua média de descarte de pacotes. Já com o tamanho do pacote em 1500 bytes (Figura 5.7(b)), há melhora na média de descarte de pacotes, mesmo essa melhora, o TCP Vegas continua sendo o que tem melhor desempenho com média de 0.5 pacotes com os nós movendo-se a 2 m/s e 3 pacotes a 12 m/s contra o TCP Reno e Vegas que possuem desempenho semelhante de 30 pacotes descartados em 2 m/s, já em 12 m/s o TCP Reno têm possui diferença de 12 pacotes descartados, sendo assim o TCP New Jersey foi o que obteve o pior resultado.

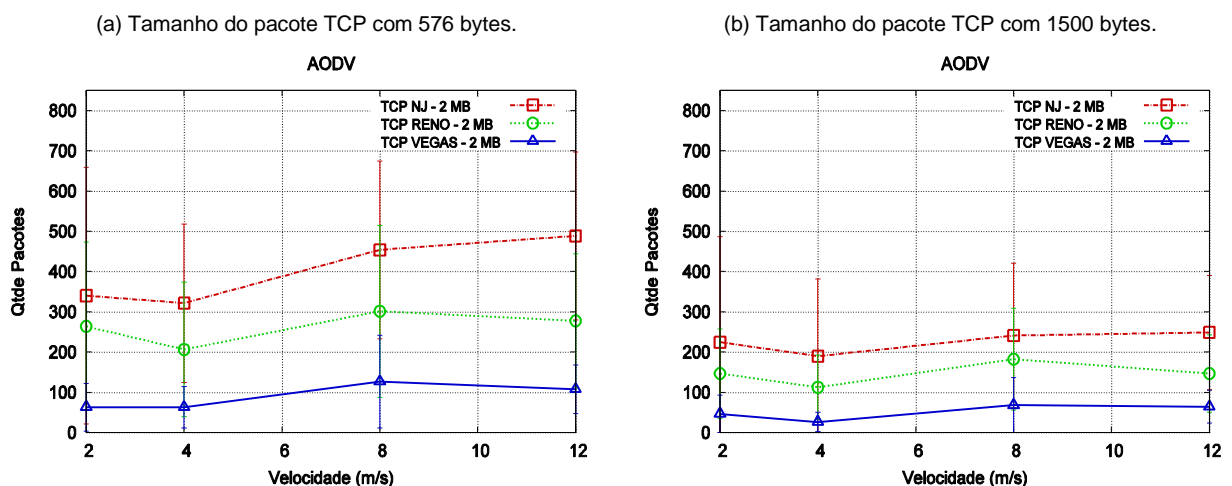
Na Figura 5.8 é apresentada os resultados considerando 5 conexões para o AODV, onde se pode observar que a média de descarte de pacotes aumenta, devido a maior quantidade de dados trafegando pela rede.

Figura 5.7: Descarte de pacotes, considerando 1 conexão para o AODV.



O TCP Vegas continua possuindo o melhor desempenho, o mesmo acontece com o TCP New Jersey que possui o pior desempenho. A média de descarte de pacotes que o TCP Vegas possui, com o pacote TCP definido em 576 (Figura 5.8(a)) é de 63 pacotes descartados, já o TCP New Jersey possui 488 pacotes descartados com os nós movendo-se a 12 m/s. A Figura 5.8(b) demonstra que com o tamanho do pacote definido em 1500 bytes as implementações melhoram suas médias de descarte de pacotes, sendo o TCP Vegas detentor do melhor desempenho com média de 43 pacotes descartados com os nós movendo-se a 2 m/s e 63 pacotes em 12 m/s. Já o TCP New Jersey e Reno têm melhora significativa em suas médias comparadas ao tamanho do pacote definido em 576 (Figura 5.8(a)), o TCP New Jersey, foi a implementação com pior desempenho possuindo uma média de 248 pacotes descartados.

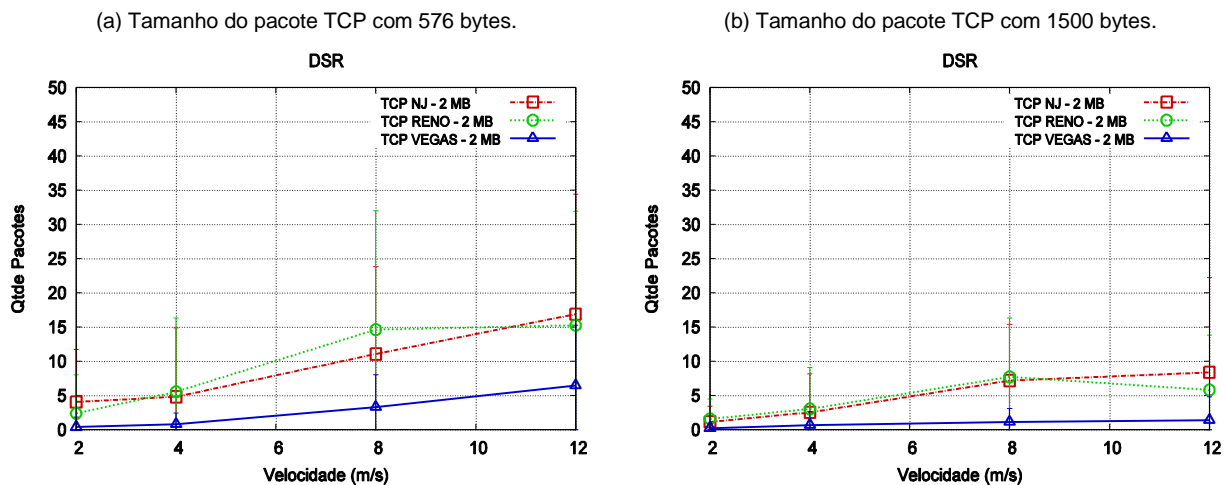
Figura 5.8: Descarte de pacotes, considerando 5 conexões simultâneas para o AODV.



Para o AODV, pode-se observar que o aumento na velocidade de movimentação dos nós, influencia o descarte de pacotes que é amenizado quando o tamanho do pacote TCP é definido em 1500 bytes. Com melhor desempenho o TCP Vegas possui as melhores médias e com as maiores médias de descarte de pacote o TCP New Jersey é o que possui o pior desempenho.

A Figura 5.9 apresenta os resultados para o descarte de pacotes, considerando 1 conexão sobre o protocolo de roteamento DSR. O TCP Vegas tem o melhor desempenho, já o TCP New Jersey e Reno possuem desempenho semelhante. Com o tamanho do pacote em 576 bytes (Figura 5.9(a)), o TCP Vegas tem uma média de 0.37 pacotes descartados com os nós movendo-se a 2m/s e 6 pacotes a 12 m/s, já o TCP New Jersey e Reno tem desempenho semelhante. Ao definir o tamanho do pacote em 1500 bytes (Figura 5.9(b)) há melhora na média de descarte de pacotes em todas as implementações, sendo o TCP Vegas o que possui melhor desempenho. O desempenho do TCP New Jersey e Reno é semelhante, mas com 12 m/s o TCP Reno se mostra melhor com uma média de 5 pacotes descartados.

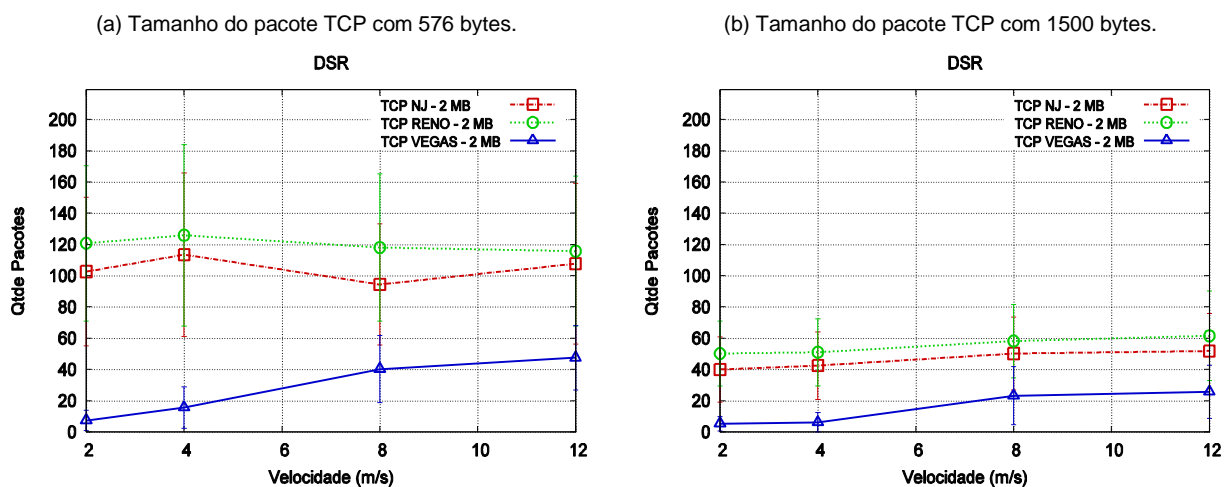
Figura 5.9: Descarte de pacotes, considerando 1 conexão para o DSR.



A Figura 5.10 considera 5 conexões sobre o DSR, o TCP Vegas tem o melhor desempenho, já o TCP Reno é que possui o pior desempenho. O TCP Vegas possui a média de 7 pacotes descartados a 2 m/s e 47 pacotes com 12 m/s, demonstrando um aumento gradativo no descarte de pacotes quando o tamanho do pacote é 576 bytes (Figura 5.10(a)), o TCP Reno é o que obteve o pior desempenho, com a média de 120 pacotes descartados em 2 m/s e 115 pacotes com os nós movendo-se 12

m/s. Comparando com o tamanho do pacote em 576 bytes (Figura 5.10(a)) é possível identificar a melhora de desempenho que as implementações possuem com o tamanho do pacote definido em 1500 bytes (Figura 5.10(b)). O TCP Vegas foi o que obteve melhor desempenho com uma média de 5 pacotes descartados com os nós movendo-se a 2 m/s e 25 pacotes em 12 m/s, tendo aumento da média em velocidades de movimentação acima de 4m/s. Com pior desempenho o TCP Reno possui média de 39 pacotes descartados a uma velocidade de 2 m/s e 51 pacotes à 12 m/s.

Figura 5.10: Descarte de pacotes, considerando 5 conexões simultâneas para o DSR.

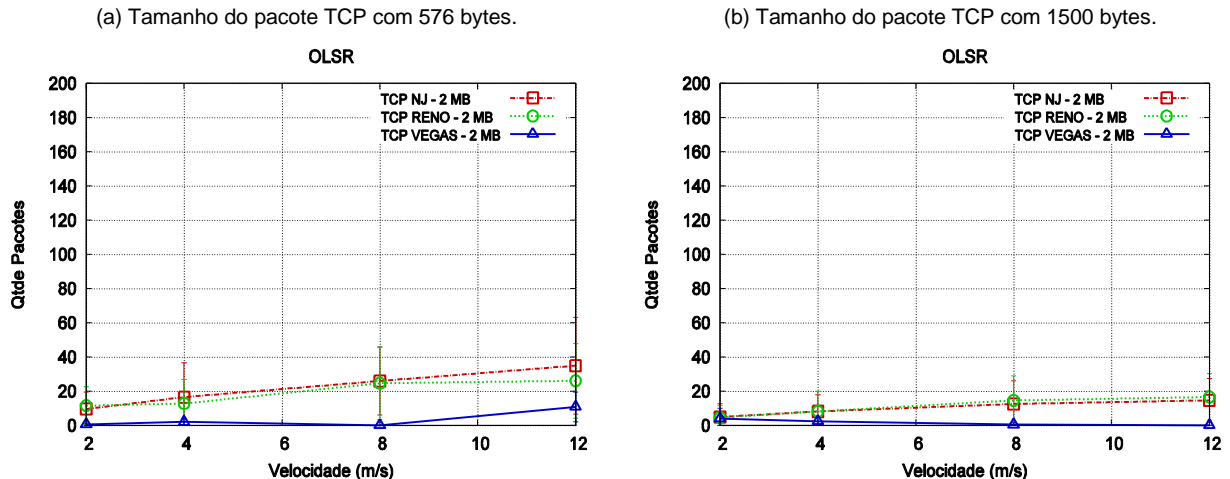


O desempenho das implementações sobre o DSR é afetado pela mobilidade dos nós e pela quantidade de pacotes de dados transmitidos pela rede, observou-se que com somente 1 conexão o TCP New Jersey e Reno possui desempenho semelhante, já com 5 conexões o TCP New Jersey tem desempenho superior ao TCP Reno, o melhor desempenho foi obtido pelo TCP Vegas. Também é possível observar que com a definição do tamanho do pacote em 1500 bytes, a média de descarte de pacotes é melhor do que com o tamanho de 576 bytes.

A Figura 5.11 demonstra os resultados de 1 conexão sobre o OLSR, o TCP Vegas possui o melhor desempenho, já o TCP New Jersey e Reno possui desempenho semelhante. O TCP Vegas obteve a melhor média de descarte de pacotes com o tamanho em 576 bytes (Figura 5.11(a)), sendo de 0.4 pacotes descartados em 2 m/s e 10 pacotes à 12 m/s, observou-se também que em velocidades de movimentação acima de 8 m/s o TCP Vegas tende a aumentar o número de pacotes descartados, já o TCP New Jersey e Reno possuem desempenho semelhante. Onde o tamanho do pacote TCP é definido em 1500 bytes

(Figura 5.11(b)), o desempenho de todas as implementações tornam-se equivalentes na velocidade de movimentação de 2 m/s, tendo uma média de 5 pacotes descartados, mas com o aumento da mobilidade dos nós o TCP Vegas mostra-se superior com uma média de 0.4 pacotes descartados em 12 m/s, já o TCP New Jersey e Reno tem desempenho semelhante.

Figura 5.11: Descarte de pacotes, considerando 1 conexão para o OLSR.

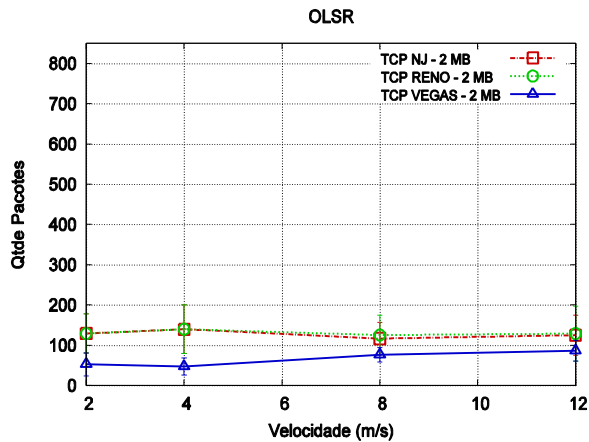


A Figura 5.12 possui 5 conexões sobre o OLSR, observa-se que o TCP Vegas tem o melhor desempenho e o TCP New Jersey e Reno possuem desempenho semelhante. Utilizando o tamanho do pacote em 576 bytes (Figura 5.12(a)), o TCP Vegas tem uma média de 50 pacotes descartados com os nós movendo-se a 2 m/s e 85 pacotes com 12 m/s, já o desempenho do TCP New Jersey e Reno é semelhante. Com o pacote definido em 1500 bytes (Figura 5.12(b)), o TCP o TCP Vegas é a implementação com melhor desempenho, sendo que na velocidade de movimentação de 8m/s o desempenho de todas as implementações são equivalentes. O desempenho das implementações no protocolo OLSR, é degradado com a alta mobilidade dos nós, com tudo foi observado que com a definição do pacote TCP em 1500 bytes a média obtida é menor do que a média do tamanho de 576 bytes.

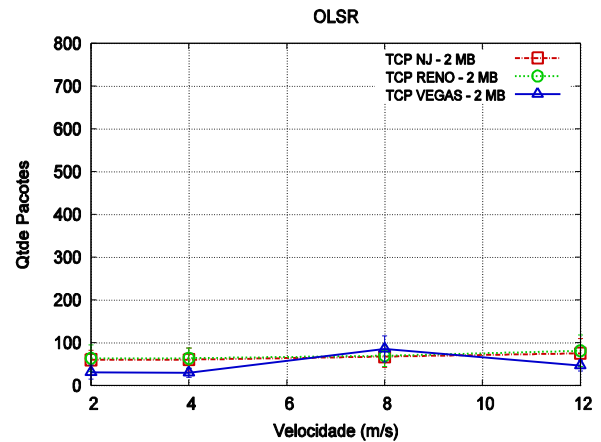
Com o melhor desempenho na métrica de descarte de pacotes o TCP Vegas obteve as melhores médias em conjunto com o OLSR. Com o aumento do número de conexões simultâneas para 5, o número de segmentos transmitidos pela rede aumento, por esse motivo a taxa de descarte obtida pelos nós nesse cenário foi maior. Outro fator que influenciou no descarte de pacotes foi a alta mobilidade dos nós, causando quebra de rotas.

Figura 5.12: Descarte de pacotes, considerando 5 conexões simultâneas para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.



5.3 Vazão efetiva

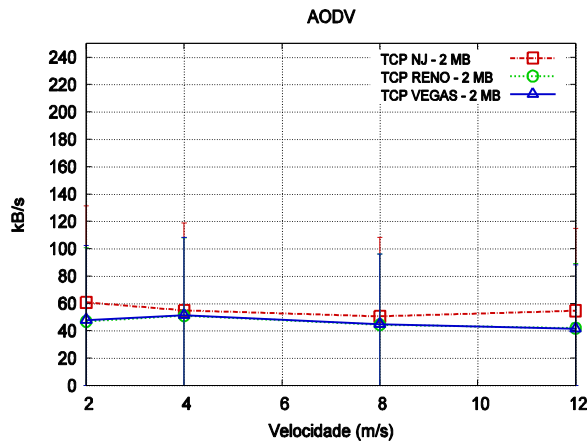
Nesta seção é apresentada a avaliação dos protocolos de roteamento sobre as diferentes implementações do TCP utilizadas por este trabalho, através da métrica de vazão efetiva.

A Figura 5.13 demonstra os resultados de somente 1 conexão sobre o AODV. O TCP New Jersey possui o melhor desempenho, já o TCP Reno e Vegas possuem desempenho semelhantes. Com o tamanho do pacote definido em 576 bytes (Figura 5.13(a)), a média de vazão do TCP New Jersey é de 60 kB/s com os nós movendo-se a 2 m/s e 54 kB/s à 12 m/s. Já o TCP Reno e Vegas possuem desempenho semelhante. O tamanho de 1500 bytes (Figura 5.13(b)), faz com que a média de vazão do TCP New Jersey aumente para 105 kB/s com os nós movendo-se a 2 m/s e 95 kB/s em 12 m/s, já o TCP Reno e Vegas possuem desempenho similar.

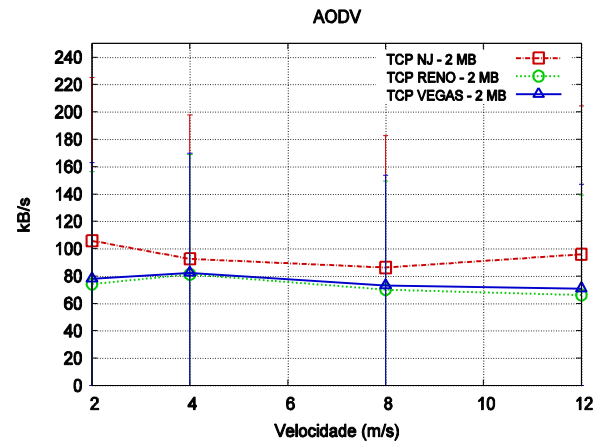
Na Figura 5.14 é considerado 5 conexões sobre o AODV, onde todas as implementações do TCP possuem desempenho semelhantes. Com o tamanho definido em 576 bytes (Figura 5.13(a)), as implementações possuem desempenho semelhante, com uma média de 63 kB/s com os nós movendo-se a 2 m/s e 107 kB/s em 12 m/s. Já com o tamanho de 1500 bytes (Figura 5.13(a)), houve aumento na média de vazão, sendo o TCP New Jersey a implementação com a melhor média de vazão em torno 43 kB/s com os nós movendo-se a 2 m/s e com resultado equivalente em 12 m/s, mantendo sua média de vazão em baixas e altas velocidades de movimentação, já o TCP Reno e Vegas possuem desempenho equivalente.

Figura 5.13: Vazão efetiva, considerando 1 conexão para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



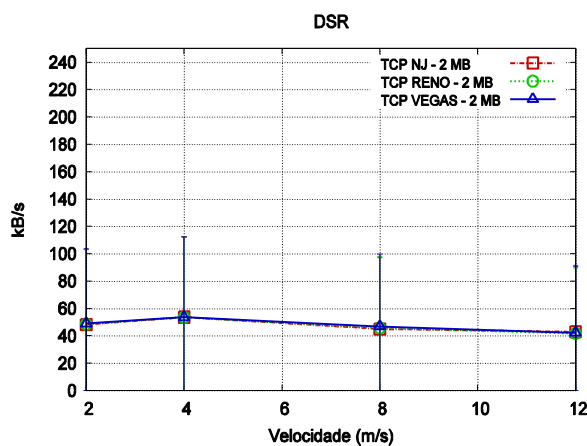
(b) Tamanho do pacote TCP com 1500 bytes.



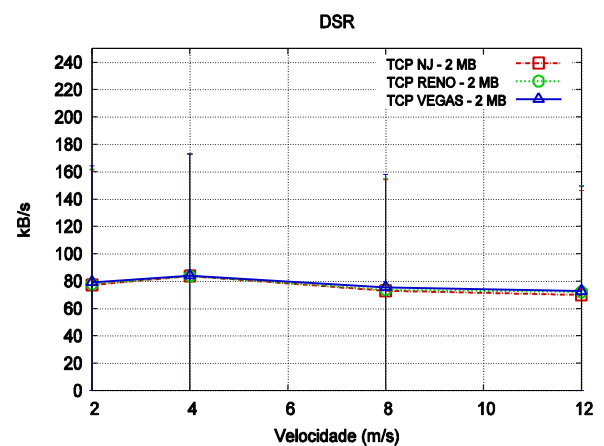
Com somente 1 conexão de dados, os resultados obtidos com o AODV são melhores do que os resultados obtidos com 5 conexões simultâneas. O TCP New Jersey, possui as melhores taxas de entrega, enquanto o TCP Reno e Vegas possuem desempenho semelhante. Observou-se também que com o tamanho do pacote em 1500 bytes, o desempenho de todas as implementações sofre degradação de desempenho.

Figura 5.14: Vazão efetiva, considerando 5 conexões simultâneas para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.

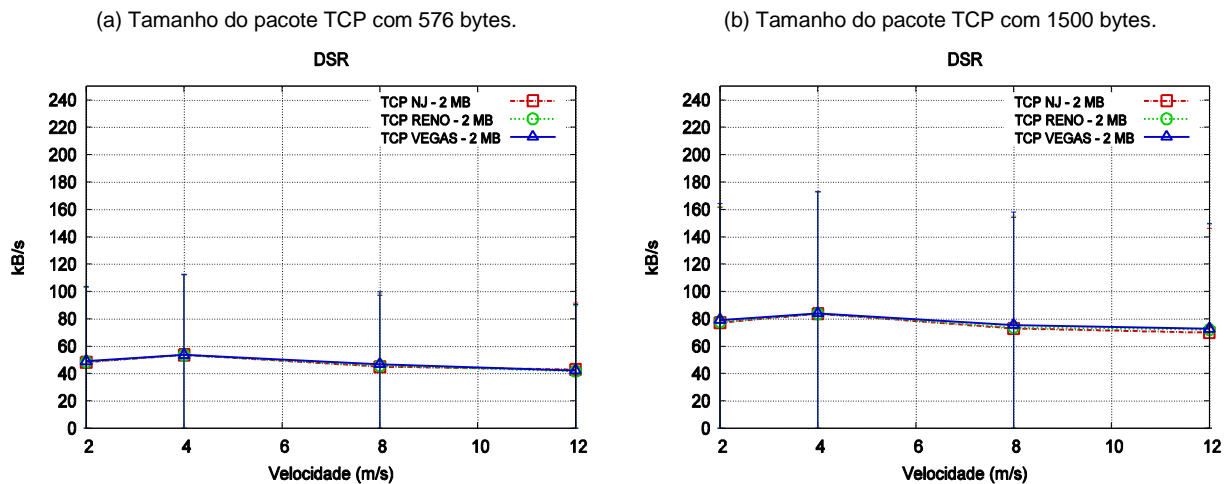


(b) Tamanho do pacote TCP com 1500 bytes.



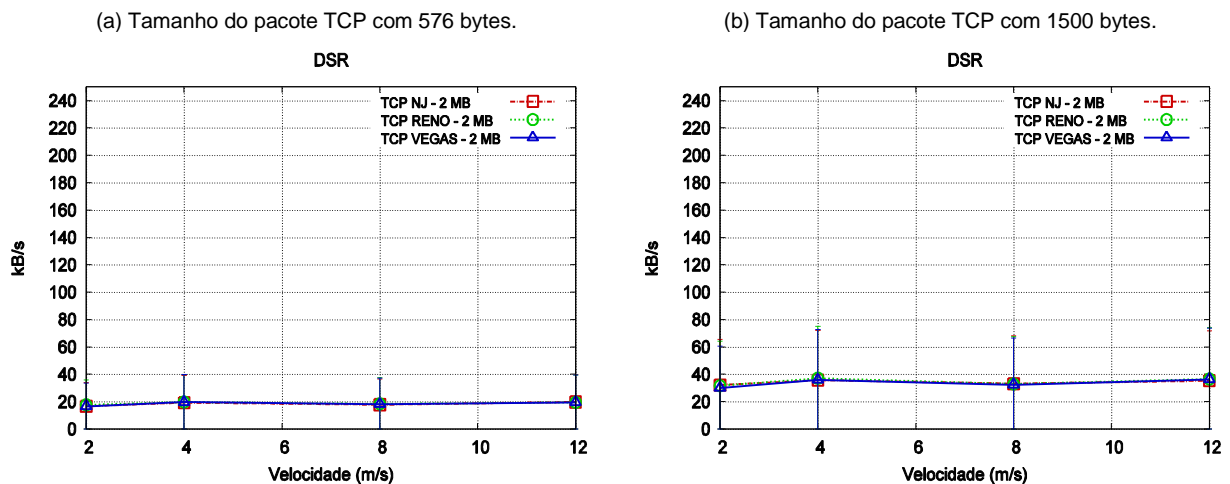
A Figura 5.15 demonstra os resultados obtidos sobre o DSR, considerando 1 conexão, neste cenário o desempenho das implementações do TCP é semelhante. Com o tamanho do pacote em 576 bytes (Figura 5.15(a)) a média das implementações é de 40 kB/s com a movimentação em 12m/s. Já com o tamanho do pacote de 1500 bytes (Figura 5.15(b)), é possível visualizar o aumento da taxa de vazão, para 70 kB/s com os nós movendo-se a 12 m/s.

Figura 5.15: Vazão efetiva, considerando 1 conexão para o DSR.



Na Figura 5.16 são consideradas 5 conexões simultâneas sobre o DSR, onde todas as implementações do TCP também possuem desempenho semelhante. Com o tamanho do pacote em 576 bytes (Figura 5.16(a)), a média obtida pelas implementações foi de 20 kB/s com os nós movendo-se a 2 m/s. Já com o tamanho do pacote em 1500 bytes (Figura 5.16(b)), o desempenho das simulações sofre uma melhora, com média de vazão de 30 kB/s com os nós movendo-se a 2 m/s.

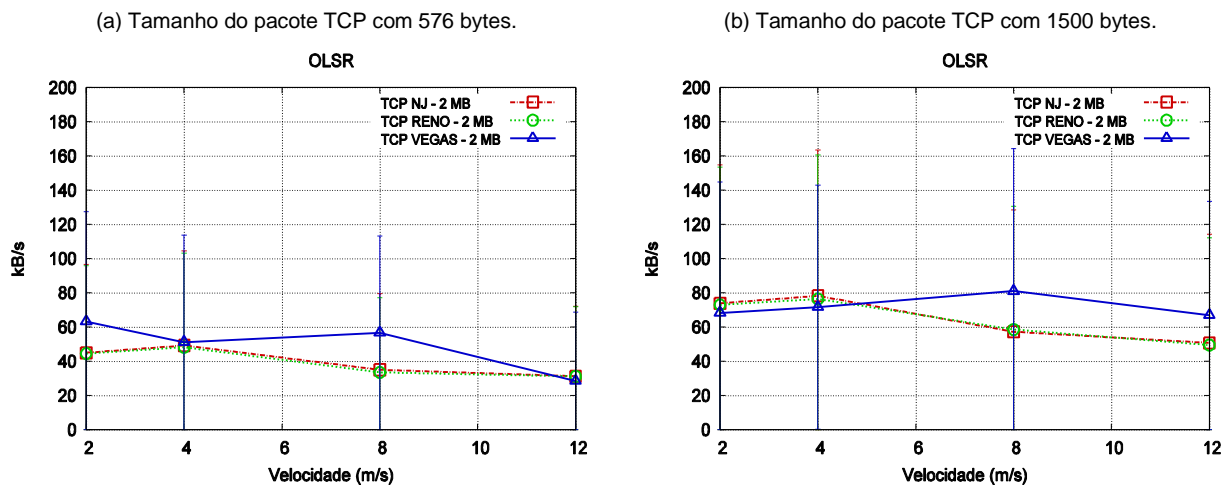
Figura 5.16: Vazão efetiva, considerando 5 conexões simultâneas para o DSR.



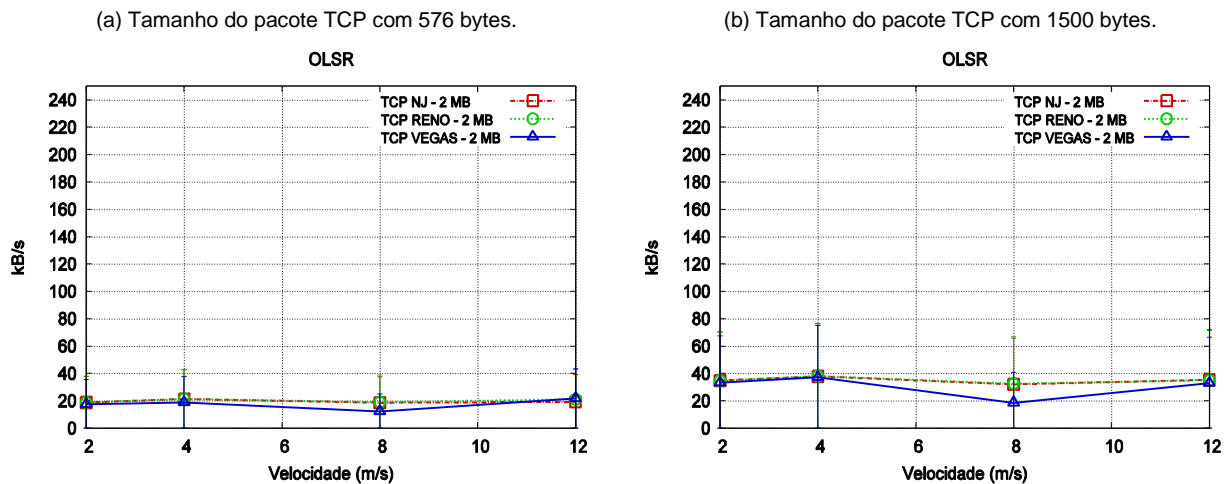
Ao comparar as diferentes implementações do TCP com o DSR, observou-se que com o tamanho do pacote TCP definido em 1500 a média de vazão efetiva obtida pelas implementações do TCP aumenta significativamente, mas é degradada com a alta mobilidade dos nós. Também foi observado que todas as implementações do TCP têm resultados equivalentes.

A Figura 5.17 demonstra os resultados considerando somente 1 conexão sobre o OLSR, onde o TCP Vegas é que possui o melhor desempenho. Com o tamanho do pacote em 576 bytes (Figura 5.17(a)), a média do TCP Vegas é de 63kB/s com os nós movendo-se a 2 m/s, mas em 12 m/s acaba possuindo desempenho semelhante ao TCP New Jersey e Reno com cerca de 30 kB/s, já o TCP New Jersey e Reno têm desempenho semelhante. Ao definir o tamanho do pacote em 1500 bytes (Figura 5.17(b)), observa-se que em 2 m/s as implementações do TCP têm desempenho semelhante até 4m/s, com o aumento da mobilidade dos nós o TCP Vegas obtêm as melhores médias com 80 kB/s a 8 m/s e 65 kB/s em 12 m/s. Já o TCP New Jersey e Reno possuem desempenho semelhante.

Figura 5.17: Vazão efetiva, considerando 1 conexão para o OLSR.



Na Figura 5.18 é exibida o desempenho do TCP sobre o OLSR, considerando 5 conexões, nesse cenário as implementações possuem desempenho semelhante. Com o tamanho do pacote em 576 bytes (Figura 5.18(a)), a média é de 20 kB/s com os nós movendo-se a 2 m/s. Já com o tamanho do pacote TCP em 1500 bytes (Figura 5.18(b)), é possível visualizar que houve aumento na média da vazão efetiva das implementações do TCP, sendo de 40 kB/s a uma velocidade de 2 m/s e mantendo em 12 m/s.

Figura 5.18: Vazão efetiva, considerando 5 conexão para o OLSR.

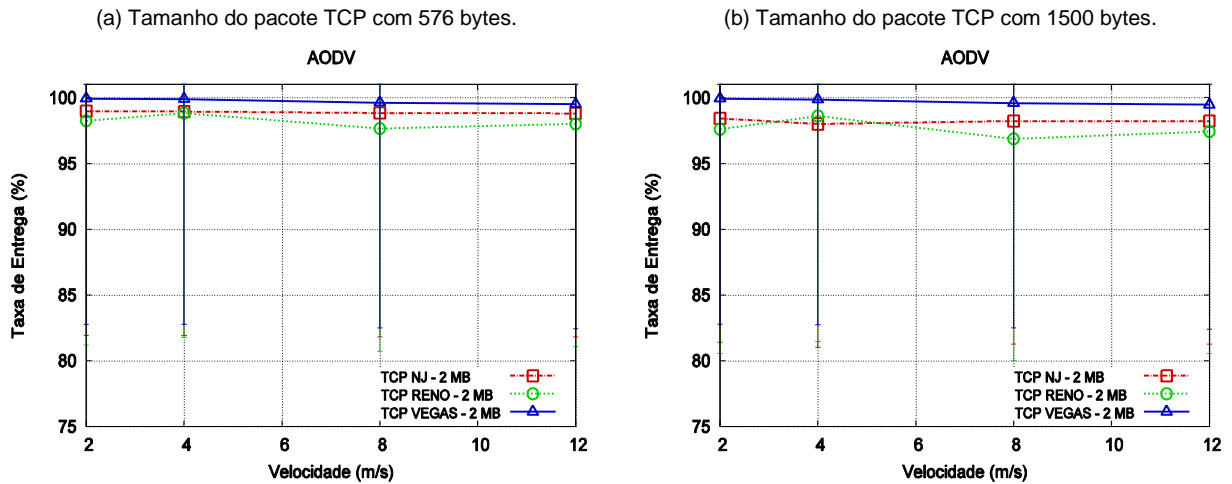
No geral, todas as implementações têm desempenho equivalente utilizando o OLSR, mais as maiores médias de vazão obtidas são utilizando o tamanho do pacote TCP em 1500 bytes. Com o tamanho do pacote TCP maior, menor é a segmentação da carga de dados a ser transmitida pela rede.

5.4 Taxa de entrega

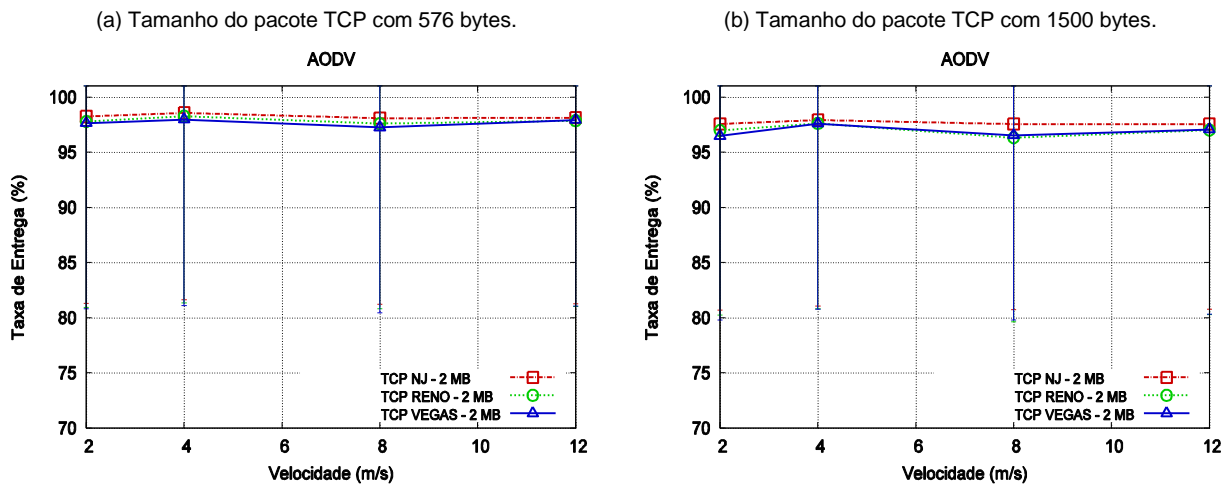
Nesta seção é apresentada a avaliação dos protocolos de roteamento sobre as diferentes implementações do TCP utilizadas por este trabalho, através da métrica de taxa de entrega.

A Figura 5.19 mostra os resultados sobre o AODV, considerando somente 1 conexão. Com uma taxa de entrega maior o TCP Vegas é que possui o melhor desempenho, já o TCP Reno tem o pior desempenho deste cenário. Com o tamanho do pacote em 576 bytes (Figura 5.19(a)), o TCP Vegas tem uma taxa de entrega de 99.9%, já o TCP New Jersey e Reno possui uma taxa de 98%. Ao definir o tamanho do pacote em 1500 bytes (Figura 5.19(b)), o TCP Vegas mantém sua taxa de entrega de 99.9%, já o TCP New Jersey e Reno sofrem perda de desempenho, tendo uma média de 97.4%.

Na Figura 5.20 o número de conexões é de 5 sobre o AODV, com este cenário o TCP New Jersey possui o melhor desempenho, já o TCP Reno e Vegas tem desempenho semelhante. Definindo o tamanho do pacote em 576 bytes (Figura 5.20(a)), o TCP New Jersey tem média de 98%, mas em 12m/s tem a mesma média de taxa de entrega que as outras implementações do TCP.

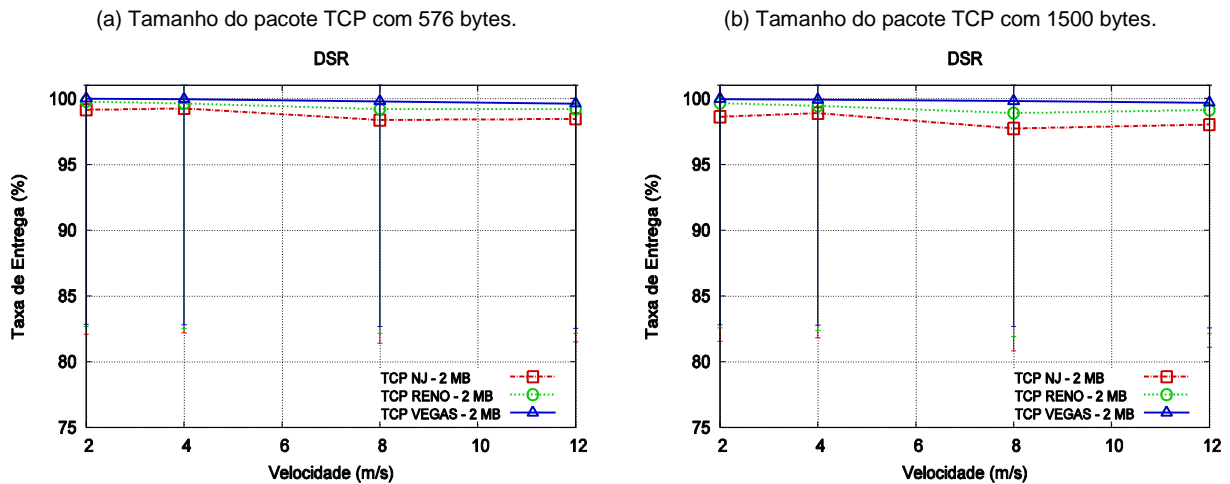
Figura 5.19: Taxa de entrega, considerando 1 conexão para o AODV.

Com os resultados obtidos sobre o protocolo de roteamento AODV, é possível observar que em cenários com baixo tráfego de dados o TCP Vegas é o que possui a melhor taxa de entrega, já em ambientes onde se possui um tráfego maior o TCP Vegas possui desempenho similar ao TCP New Jersey e Reno.

Figura 5.20: Taxa de entrega, considerando 5 conexões simultâneas para o AODV.

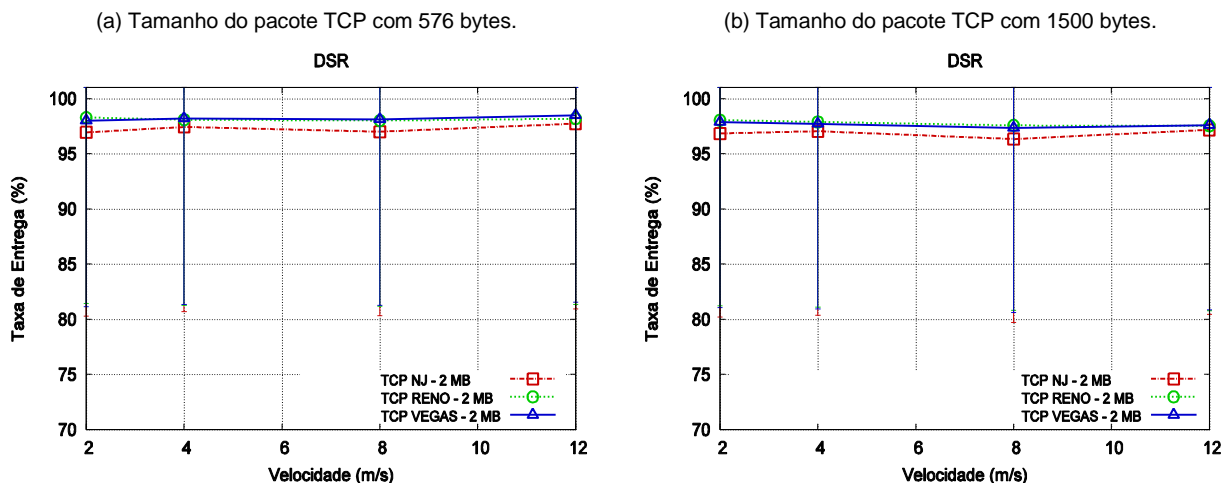
A Figura 5.21 mostra os resultados das simulações sobre o DSR, considerando somente 1 conexão. Para este cenário o TCP Vegas foi que obteve o melhor desempenho, já o pior desempenho foi do TCP New Jersey. Com o tamanho do pacote em 576 bytes (Figura 5.21(a)), o TCP Vegas possui uma média de 99%. Já com o tamanho do pacote em 1500 bytes (Figura 5.21(b)), o New Jersey possui o pior desempenho, possuindo uma média de taxa de entrega de 98%. O TCP New Jersey é a implementação que mais sofreu degradação de desempenho, devido a mobilidade dos nós.

Figura 5.21: Taxa de entrega, considerando 1 conexão para o DSR.



Na Figura 5.22 é apresentada os resultados sobre o DSR, considerando 5 conexões. Neste cenário o TCP Reno e Vegas possuem desempenho semelhante, já o TCP New Jersey é que possui o pior desempenho. Com o tamanho do pacote em 576 bytes (Figura 5.22(a)), o TCP New Jersey possui a média de 97% com os nós movendo-se a 2m/s, já o TCP Reno e Vegas possui uma taxa de entrega de 98%. Utilizando o tamanho do pacote em 1500 bytes (Figura 5.22(b)), o TCP New Jersey possui uma média de 96% de taxa de entrega, já o TCP New Jersey e Reno possuem desempenho semelhante a 97%.

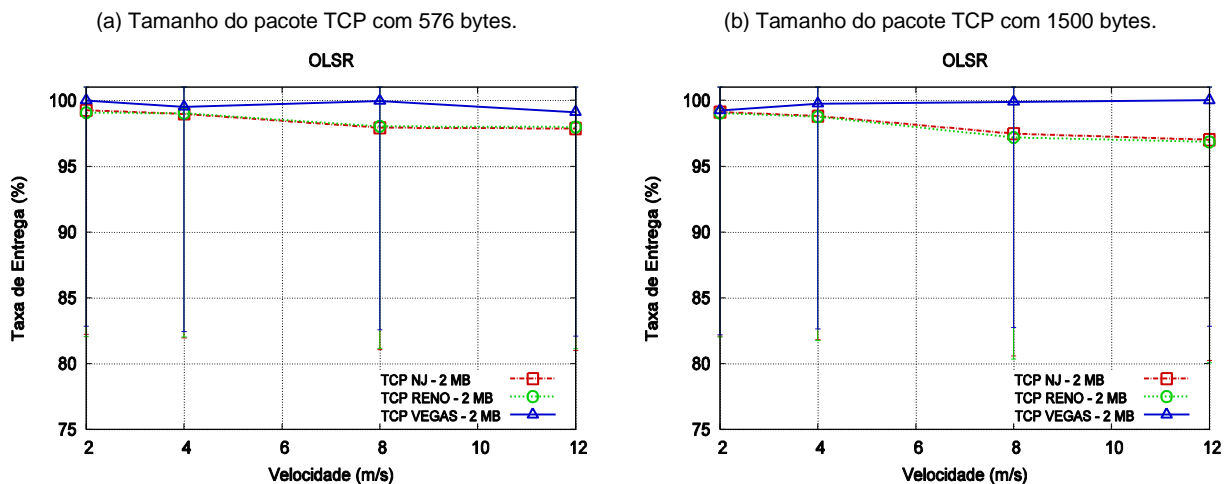
Figura 5.22: Taxa de entrega, considerando 5 conexões simultâneas para o DSR.



Ao utilizar o DSR, o desempenho do TCP Vegas tem melhores resultados em ambientes com baixo tráfego de dados, já com uma carga maior de dados seu desempenho é degradado, obtendo resultados semelhantes ao TCP Reno. O TCP New Jersey sofre perdas de desempenho em velocidades altas de movimentação e maior carga de dados sendo transmitida pela rede.

A Figura 5.23 apresenta o resultado obtido sobre o OLSR, considerando somente 1 conexão. Neste ambiente o protocolo que tem melhor desempenho é o TCP Vegas, já o TCP New Jersey e Reno possuem desempenho semelhante. Com o tamanho do pacote em 576 bytes (Figura 5.23(a)), o TCP Vegas possui uma média de 99% de taxa de entrega, já o TCP New Jersey e Reno tem desempenho semelhante a 99%. Ao utilizar o tamanho do pacote em 1500 bytes (Figura 5.23(b)), o TCP Vegas tem uma melhora em altas velocidades de movimentação, chegando a uma média de 100% de taxa de entrega, já o TCP New Jersey e Reno começam com uma taxa de 99% a 2m/s mais com o aumento da mobilidade dos nós chega a 96% em 12 m/s.

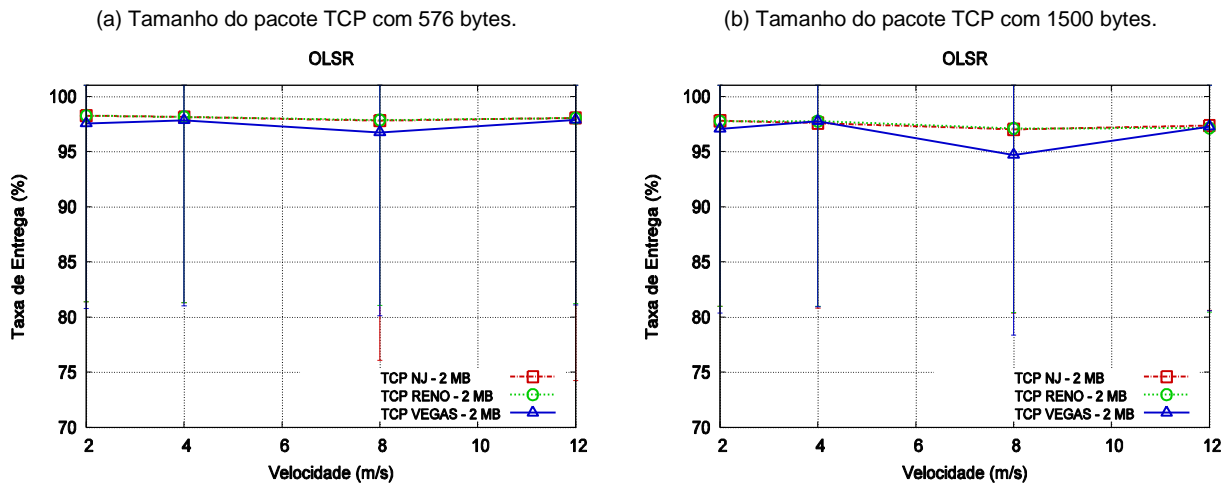
Figura 5.23: Taxa de entrega, considerando 1 conexão para o OLSR.



A Figura 5.24 demonstra os resultados considerando 5 conexões utilizadas sobre o OLSR. Para este ambiente as implementações do TCP possuem o mesmo desempenho. Com o tamanho do pacote em 576 bytes (Figura 5.24(a)), é possível observar que a média obtida é de 97%. Já com o uso do tamanho do pacote em 1500 bytes (Figura 5.24(b)), o desempenho do TCP Vegas é pior em 8m/s com 95% da taxa de entrega, seu desempenho melhora novamente quando a velocidade alcança 12m/s.

Ao utilizar o OLSR, o TCP Vegas tem melhor desempenho com 1 conexão e tamanho do pacote TCP definido em 1500 bytes, já neste ambiente o TCP New Jersey e Reno possuem desempenho semelhante. Com o aumento do número de conexões para 5, o desempenho do TCP Vegas sofre degradação, já o TCP New Jersey e Reno se mantém com desempenho semelhante.

Figura 5.24: Taxa de entrega, considerando 5 conexões simultâneas para o OLSR.



O comportamento das implementações do TCP sobre o protocolo de roteamento das MANETS se mostrou melhor com o uso de somente 1 conexão, sendo o TCP Vegas o que possui melhor desempenho, mas em ambientes onde se tem alta taxa de dados trafegando pela rede, o TCP Vegas tem seu desempenho degradado, sendo pior que o TCP New Jersey e Reno que se mantém na média de 97% de taxa de entrega.

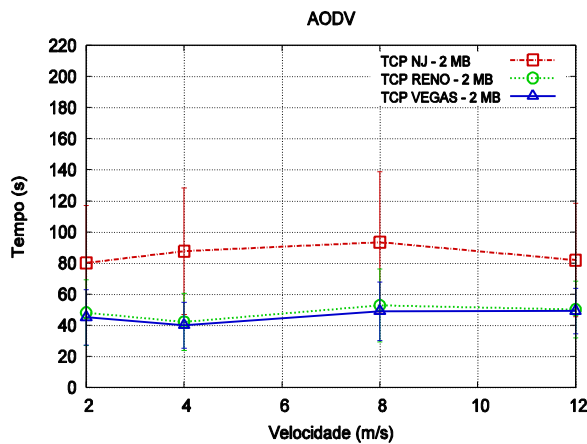
5.5 Tempo total de envio de dados

Nesta seção é apresentada a avaliação dos protocolos de roteamento sobre as diferentes implementações do TCP utilizadas por este trabalho, através da métrica de tempo total de envio de dados.

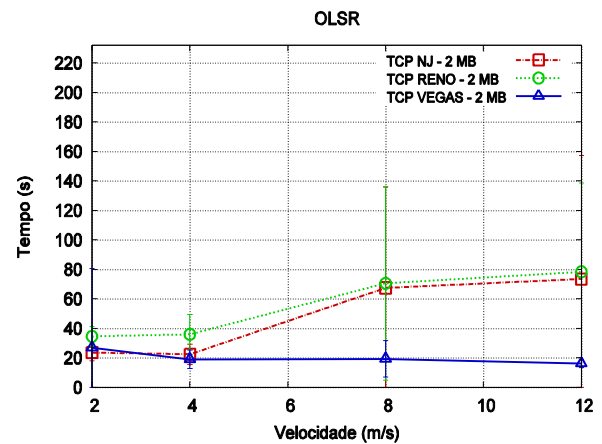
A Figura 5.25 apresenta as médias do tempo de envio de dados, considerando 1 conexão sobre o AODV. Para este ambiente o TCP New Jersey é que possui o pior desempenho, já o TCP Reno e Vegas possuem desempenho semelhante. Com o tamanho do pacote definido em 576 bytes (Figura 5.25(a)), o TCP Reno e Vegas possuem um tempo médio de 45s com os nós movendo-se a 2 m/s e 50s à 12 m/s, já o TCP New Jersey tem o tempo médio de 80s. Ao utilizar o tamanho do pacote em 1500 bytes (Figura 5.25(b)), o TCP Vegas possui uma média de 20s com os nós movendo-se a 2 m/s e 22s em 12 m/s, já o TCP Reno e Vegas tem uma média de 24s com movimentação de 2 m/s e 27s com 12 m/s.

Figura 5.25: Tempo total de envio de dados, considerando 1 conexão para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



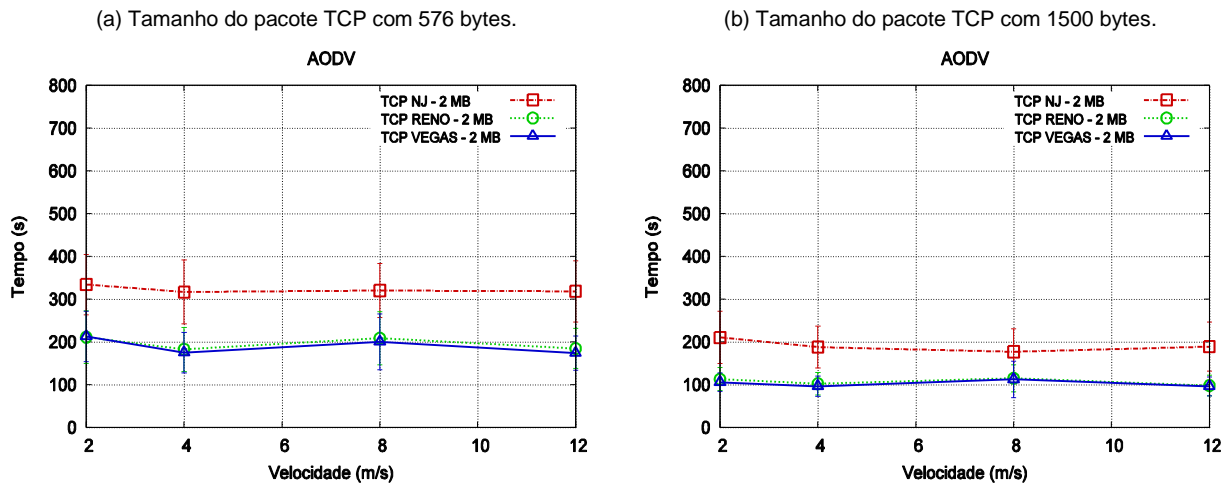
(b) Tamanho do pacote TCP com 1500 bytes.



Na Figura 5.26 é considerado 5 conexões utilizadas sobre o AODV. Neste ambiente o TCP New Jersey possui o pior desempenho, já o TCP Reno e Vegas tem desempenho semelhante. Com o uso do tamanho do pacote em 576 bytes (Figura 5.26(a)), o TCP Reno e Vegas tem uma média de 212s com os nós movendo-se a 2 m/s e 185s a uma velocidade de 12 m/s, já o TCP New Jersey possui uma média de 335s com movimentação de 2 m/s e 315s à 12 m/s. Ao utilizar o tamanho do pacote em 1500 bytes (Figura 5.26(b)), o TCP Reno e Vegas possuem uma média de 105s com os nós movendo-se a 2 m/s e 95s à 12 m/s, já o TCP New Jersey tem uma média de 210s com velocidade de movimentação de 2 m/s e 185s à 12 m/s.

Ao analisar os resultados gerados pelas implementações do TCP utilizadas sobre o protocolo de roteamento das MANETs o AODV, pode-se observar que com o tamanho do pacote TCP definido em 576 bytes têm-se as maiores médias de tempo, já com a definição do tamanho do pacote TCP em 1500, houve melhora na média de tempo utilizada para transmitir todos os dados.

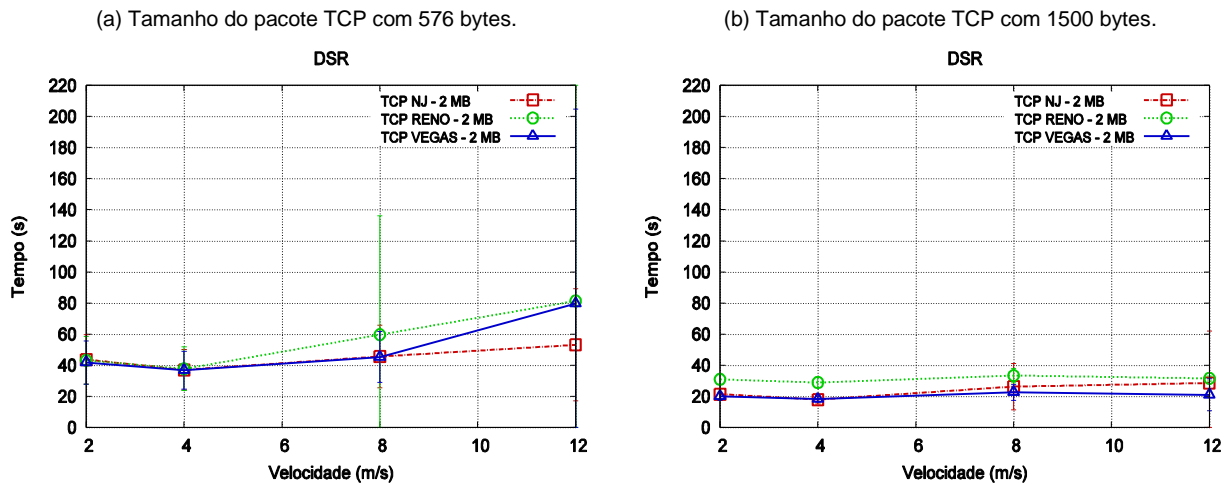
Figura 5.26: Tempo total de envio de dados, considerando 5 conexões simultâneas para o AODV.



A Figura 5.27 apresenta os resultados sobre o DSR, considerando somente 1 conexão. Neste ambiente o TCP Reno apresenta o melhor desempenho e com pior desempenho encontra o TCP Vegas. Com o tamanho do pacote em 576 bytes (Figura 5.27(a)), o TCP New Jersey tem uma média de 43s com os nós movendo-se a 2m/s e 53s à 12 m/s, mas também é possível ver que o TCP Vegas possui desempenho semelhante ao TCP New Jersey até os 8m/s de movimentação, tendo desempenho equivalente ao TCP Reno em 12/s com uma média de 80s, o mesmo acontece com o TCP Reno. Ao utilizar o tamanho do pacote em 1500 bytes (Figura 5.27(b)), o TCP Vegas mantém uma média de 20s com nós movendo-se a 2 m/s e 28s com 12 m/s, mas observa-se também que o TCP New Jersey possui desempenho semelhante ao TCP Vegas com velocidades abaixo de 4 m/s, acima desta velocidade seu desempenho é degradado, possuindo a 12 m/s média equivalente ao TCP Reno que obteve uma média de 30s.

Na Figura 5.28 utiliza-se 5 conexões sobre o DSR. Com o tamanho do pacote em 576 bytes (Figura 5.28(a)), o TCP New Jersey apresenta melhor desempenho, com uma média de 200s com os nós movendo-se a 2 m/s e 230s à 12 m/s, já o TCP Vegas e Reno têm desempenho semelhante ao TCP New Jersey com a movimentação em 2m/s, mas com o aumento da mobilidade dos nós o TCP Reno é o que tem o pior desempenho com uma média de 450s em 12 m/s.

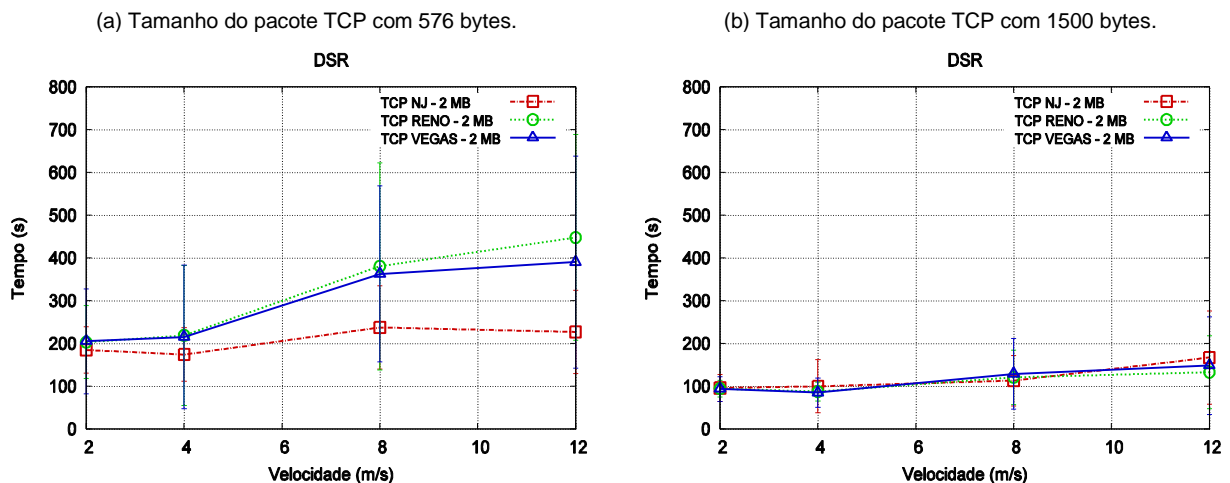
Figura 5.27: Tempo total de envio de dados, considerando 1 conexão para o DSR.



Ao utilizar o tamanho em 1500 bytes (Figura 5.28(b)), o TCP New Jersey, Reno e Vegas tem desempenho semelhante com uma média de 100s com os nós movendo-se a 2 m/s e 160s à 12 m/s.

Ao observar a variação de velocidade de movimentação é possível ver que quanto maior a velocidade de movimentação dos nós, maior é o tempo de envio de dados que as implementações do TCP utilizadas sobre o DSR necessitam para encaminhar os pacotes de dados. Observou-se também que com a definição do tamanho do pacote TCP em 1500 bytes, o tempo para envio de dados é menor comparado ao tamanho de pacote definido em 576 bytes.

Figura 5.28: Tempo total de envio de dados, considerando 5 conexões simultâneas para o DSR.



A Figura 5.29 apresenta os resultados obtidos sobre o OLSR, considerando somente 1 conexão. Com o tamanho do pacote em 576 bytes (Figura 5.29(a)), o TCP New Jersey possui o melhor desempenho até 8m/s, com uma média de 48s para envio da página, em velocidades acima de 8m/s o TCP Vegas possui o melhor

desempenho com uma média de 70s com os nós movendo-se a 12 m/s. Já TCP Reno é o que possui o desempenho mais afetado pela alta mobilidade dos nós, com uma média de 51s com movimentação de 2 m/s e 113s em 12m/s. Ao utilizar o tamanho do pacote em 1500 bytes (Figura 5.29(b)), o TCP Vegas é que possui o melhor desempenho, possuindo uma média de tempo de 26s com os nós movendo-se a 2 m/s e 16s à 12 m/s, já o TCP Reno possui o pior desempenho com uma média de 23s com os nós movendo-se a 2 m/s e 73s à 12 m/s.

Figura 5.29: Tempo total de envio de dados, considerando 1 conexão para o OLSR.

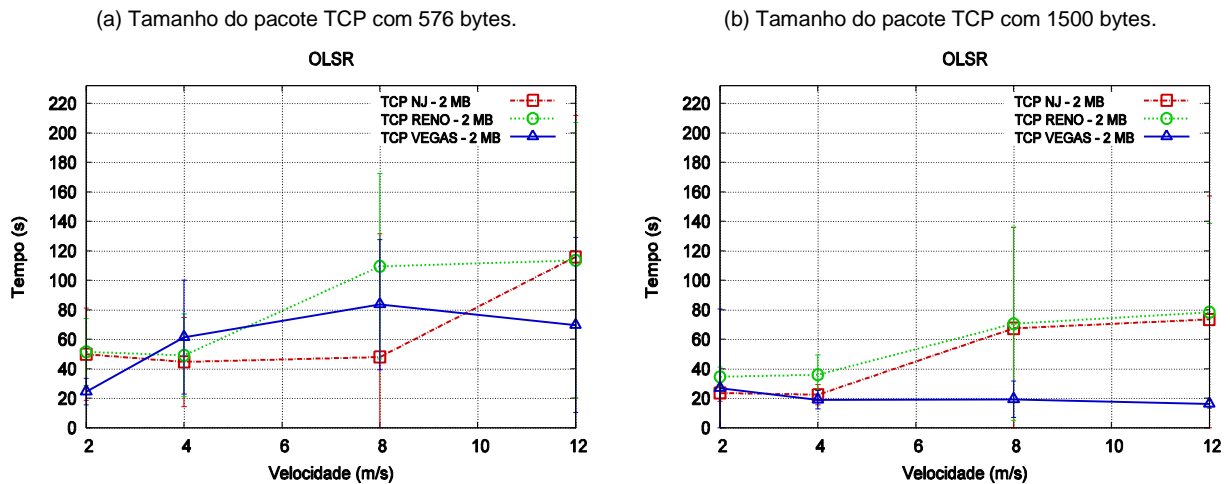
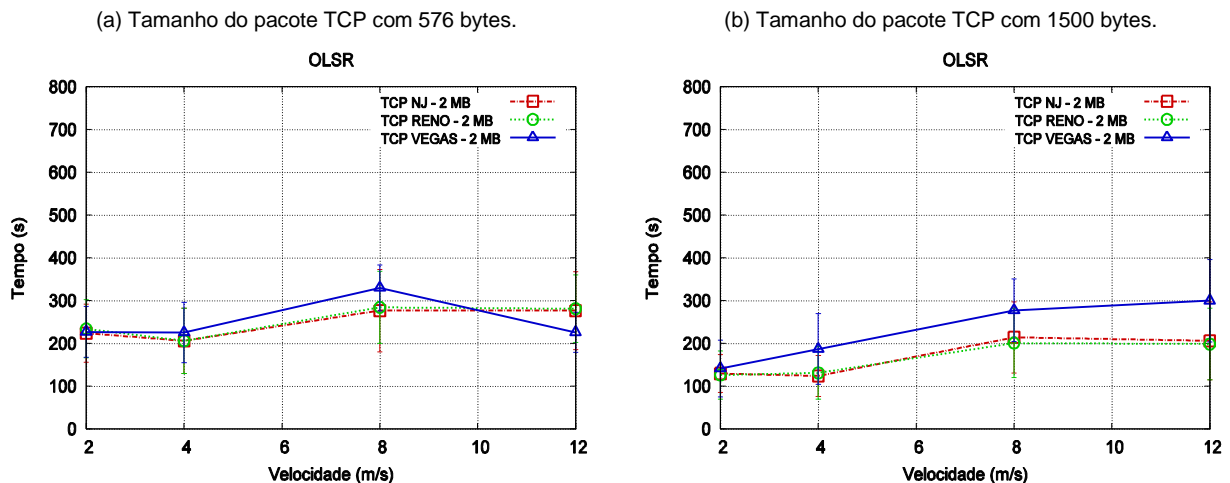


Figura 5.30: Tempo total de envio de dados, considerando 5 conexões simultâneas para o OLSR.



Na Figura 5.30 são apresentados os resultados com 5 conexões utilizando o protocolo de roteamento OLSR. Na transmissão de pacotes de 576 bytes (Figura 5.30(a)), é possível visualizar que o desempenho do TCP Vegas, sofre perda de desempenho com a velocidade de 8 m/s. Com os nós movendo-se a 2 m/s sua média de tempo é de 226s, chegando a 330s com movimentação de 8 m/s e

terminando com a mesma média de 2 m/s em 12 m/s. já o TCP New Jersey e Reno possuem desempenho semelhante com uma média de tempo de 233s com os nós movendo-se a 2 m/s e 280s à 12 m/s, possuindo uma pequena degradação de desempenho com a alta mobilidade dos nós. Utilizando o tamanho do pacote em 1500 bytes (Figura 5.30(b)), o TCP Vegas obtém o pior desempenho com uma média de 140s com os nós movendo-se a 2 m/s e 300s à 12 m/s. Já o TCP New Jersey e o TCP Reno possuem desempenho semelhante com uma média de 126s, com movimentação de 2 m/s e 200s à 12 m/s.

5.6 Considerações finais

Através da análise dos resultados, observa-se que a utilização do tamanho do pacote TCP definido em 1500 bytes, faz com que haja melhora no desempenho das implementações nas métricas de vazão efetiva, descarte de pacotes e tempo de envio de dados. O aumento do número de conexões e carga de dados transmitidos pela rede faz com que o desempenho das implementações do TCP seja degradado, pois influencia na quantidade de segmentos que é transmitido.

Tabela 5.1: Desempenho das implementações do TCP sobre o protocolo de roteamento das MANETS.

Métrica Protocolo	Taxa de Entrega	Vazão Efetiva	Atraso	Tempo Total de Envio de Dados	Descarte Total de Pacotes
Vegas + AODV	X	X	X	X	X
Vegas + DSR	X	X	X	X	X
Vegas + OLSR	X	X	X	X	X
Reno + AODV	X	X			
Reno + DSR	X	X			
Reno + OLSR	X	X			
NJ + AODV		X			
NJ + DSR		X			
NJ + OLSR		X			

X – Melhor desempenho obtido.

Na Tabela 5.1 é apresentado um resumo dos resultados alcançados neste trabalho. Por meio da Tabela 5.1 pode-se observar que o TCP Vegas é a implementação que têm os melhores resultados na métrica de atraso, descarte de

pacotes e tempo de envio de dados, com resultados semelhantes ao TCP New Jersey e Reno na métrica de vazão efetiva da rede e taxa de entrega similar ao TCP Reno.

Foi observado que com a utilização do protocolo de roteamento da MANET o AODV, o desempenho do TCP Vegas é melhor nas métricas de atraso, descarte de pacotes e taxa de entrega, possuindo desempenho semelhante na métrica de tempo de envio com o TCP Reno aplicado ao AODV.

6 CONCLUSÃO

Com a utilização de redes MANET, pode-se criar de modo fácil e rápido uma rede em locais que não dispõem de infraestrutura já definida, a fim de compartilhar recursos. Alguns dos ambientes que uma MANET pode ser empregada são aqueles ambientes que não se pode, ou não se tem condições de criar uma infraestrutura fixa. A utilização para compartilhar recursos aliada a mobilidade, faz da MANET uma grande ferramenta, onde pode-se criar de forma simples e barata uma rede no intuito de compartilhar acesso a Internet, somente conectando os dispositivos uns aos outros. E são os dispositivos que ficam responsáveis por gerenciar a rede, durante seu tempo de atividade, sem a necessidade de gastos com infraestrutura. Mas ao utilizar uma MANET como extensão da Internet, problemas com o desempenho do TCP surgem, o TCP não consegue identificar as falhas que ocorrem na rede. O TCP trata as falhas como congestionamento e solução é diminuir sua janela de transmissão, impactando em seu desempenho.

Neste trabalho foi realizada uma avaliação do uso do TCP Vegas, New Jersey e Reno sobre os protocolos de roteamento das MANETS o AODV, DSR e OLSR. As avaliações foram feitas em diferentes cenários, no intuito de abstrair para o cenário de simulação o mundo real.

Com base nos resultados obtidos no Capítulo 5, observou-se que o TCP Vegas foi a implementação do TCP que obteve o melhor desempenho quando aplicado aos protocolos de roteamento das MANETS o AODV, DSR e OLSR. O desempenho obtido pelo TCP Vegas se dá devido a sua estimativa de banda eficiente, com isso o TCP Vegas consegue gerenciar melhor as falhas que ocorrem na rede e assim não diminui sua janela de transmissão de dados desnecessariamente, em redes MANETS onde a mobilidade dos nós é alta isso é o diferencial para se obter um maior desempenho, pois com a grande movimentação dos nós pode-se ter alta taxa de quebra de rotas, fazendo com que o pacote de dados não seja entregue.

A utilização do tamanho do TCP em 1500 bytes faz com que as médias dos resultados sejam melhores, pois com o aumento do tamanho do TCP ocorre a menor segmentação da quantidade de dados transmitida, influenciando na quantidade de pacotes de dados que são transmitidos pela rede. O aumento do número de conexões sendo realizadas na rede causa maior tráfego de dados, por

este motivo as médias dos resultados para 5 conexões simultâneas foi maior do que somente para 1 conexão.

O TCP Vegas e o protocolo de roteamento da MANET o AODV, é o mais indicado para uso como extensão da Internet em cenários que possuem características semelhantes às simuladas por este trabalho, devido a seu desempenho obtido com as métricas de avaliação. A utilização do TCP Vegas é melhor quando seu tamanho de pacote é definido em 1500 bytes para cenários que possui maior tráfego de dados.

6.1 Trabalhos Futuros

Como trabalho futuro pretende-se:

- Realizar avaliações levando em consideração um maior número de nós;
- Realizar a avaliação de outras implementações do TCP como: TCP Westwood, TCP SACK entre outros;
- Realizar todas as simulações presentes nesse trabalho em um ambiente que represente uma rede densa.

REFERÊNCIAS

AHO, Alfred V; KERNIGHAN, Brian W; WEINBERGER, Peter J. **The AWK programming language**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1987.

AREAL, J. L. **Proposta de um Modelo de Confiança para o Protocolo OLSR**. 2008. 72 f. Dissertação (Mestrado em Engenharia Elétrica) – Departamento de Engenharia Elétrica – Universidade de Brasília, Brasília. 2008.

BASAGNI, S. et al. **Mobile Ad Hoc Networking**. Hoboken, N.J.: Wiley-IEEE Press, 2004.

BHAGWAT, C. P. P. **Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers**. In: ACM SIGCOMM 94, 1994, Londres. Proceedings New York: ACM Press, 1994.

CASSETTI, C; GERLA, M; MASCOLO, S; SANADIDI, M.Y. **Westwood: End-to- End Congestion Control for Wired/Wireless Networks**. Wireless Networks Journal, Vol.8. pp. 467-479, 2002.

CASSETTI, C; GERLA, M; MASCOLO, S; SANADIDI, M.Y. **TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links**. Mobicom. Rome, Italy: ACM, 2001.

CAVALCANTI, J. L. **Análise Comparativa dos Algoritmos de Controle de Congestionamento do TCP**. 2005. 59 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, 2005.

CLAUSEN, T; JACQUET, P. **Optimized Link State Routing Protocol (OLSR)**, IETF RFC-3626, 2003.

FALL, K; Floyd, S. **Simulation-Based Comparisons of Tahoe, Reno and SACK TCP**. SIGCOMM Comput. Commun. Rev., 26:5–21, 1996.

FERNANDES, B. V. **Protocolos de roteamento em redes Ad Hoc**. 2003. 74 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Computação, Universidade Estadual de Campinas, Campinas. 2003.

HENNA, S. **A Throughput Analysis of TCP Variants in Mobile Wireless Networks**. Third International Conference on Next Generation Mobile Applications, Services and Technologies, 2009.

HOLLAND, G. VAIDYA, N. **Analysis of TCP Performance over Mobile Ad Hoc Networks**. Kluwer Academic Publishers. Manufactured in The Netherlands, 2002.

JOHNSON D. B; MALTZ, D. A. **Dynamic Source Routing in Ad Hoc Wireless Networks**. Mobile Computing, páginas 153-181. Kluwer Academic Publishers, 1996.

KUROSE, J. F. e ROSS, K. - **Redes de Computadores e a Internet - 5ª Ed.**, Pearson, 2010.

LAWRENCE S; BRAKMO; PETERSON, L. L. **TCP Vegas: End to End Congestion Avoidance on a Global Internet**. IEEE Journal on selected Areas in communications, 13:1465–1480, 1995.

LIU, P. et al. **Analyzing the TCP Performance on Mobile Ad-Hoc Networks**. Korea South. 13th International Conference on Advanced Communication Technology (ICACT), 2011.

MACHADO, Javam. C.; LEONARDO, O.; SOUZA, Flávio. R.C. **Computação em nuvem: Conceitos, Tecnologias, Aplicações e Desafios**. Universidade Federal do Ceará. ERCMAPI 2009.

MO, R. et al. **Analysis and Comparison of TCP Reno and Vegas**. INFOCOM'99. Berkeley, California: IEEE, 1999.

PERKINS, C., ROYER, E. **Ad Hoc On-Demand Distance Vector Routing**. In: 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans. Proceedings... Los Alamitos: IEEE Computer Society, fev. 1999.

POSSER, L. N, **Computação Móvel e Mlearning**. Porto Alegre: UniRitter, 2006.

SEDDIK-GHALED, A., GHAMRI-DOUDANE, Y., SENOUCI, S.-M. **Effect of Ad Hoc Routing Protocols on TCP Performance within MANETs**. Sensor and Ad Hoc Communications and Networks, Reston. 2006.

MCCANNE, S e FLOYD, S. **NS Network Simulador**, acesso em 2014. <http://www.isi.edu/nsnam/ns/>

STEVENS, W. **TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms**. IETF RFC-2001, 1997.

TANENBAUM, Andrew S. **Redes de computadores**. 4. ed. Rio de Janeiro: Editora Elsevier, 2003.

XU K; TIAN, Y; ANSARI, N. **Improving TCP Performance in Integrated Wireless Communications Networks**. Computer Networks, 47(2):219–237, 2005.

XU, K; TIAN, , Y; ANSARI, N. **TCP-Jersey for Wireless IP Communications**. IEEE Journal on Selected Areas in Communications, 22(4):747–756, 2004.

APÊNDICE A

RESULTADOS DAS SIMULAÇÕES COM TRANSMISSÃO DE ARQUIVOS DE 500 KB

No Apêndice são apresentadas as avaliações do impacto das diferentes implementações do TCP sobre os protocolos de roteamento das MANETS, utilizando os parâmetros de configuração definidos na Tabela 4.1 considerando a transmissão de 500 KB.

Para demonstrar os resultados gerados através das métricas descritas no item 4.2, foi utilizado o gráfico de linha. Cada gráfico contém três cenários simulados, sendo cada linha do gráfico a representação de um cenário. As marcações verticais de cada ponto no gráfico representam o desvio padrão da métrica avaliada, obtidos através de 35 simulações realizadas para cada ponto.

Os cenários encontrados nos gráficos é o resultado obtido através da métrica avaliada com o uso do TCP New Jersey, Reno e Vegas sobre os protocolos de roteamento o AODV, DSR e OLSR, variando a quantidade de conexões em 1 e 5 para o tamanho do pacote TCP de 576 e 1500 bytes.

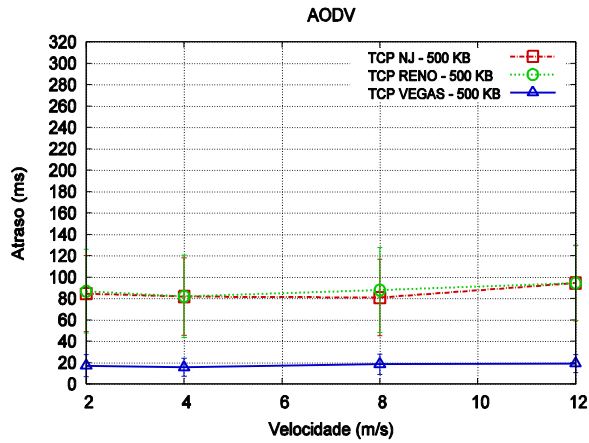
Para cada métrica avaliada foi criado um item, contendo os gráficos dispostos lado a lado para uma melhor visualização do impacto causado pela modificação do tamanho do pacote TCP sobre os protocolos de roteamento.

MÉTRICA DE ATRASO.

Nesta seção são apresentados os resultados obtidos pelas implementações do TCP sobre os protocolos de roteamento das MANETS, avaliando a métricas de atraso na entrega de pacotes.

Figura A.1: Considerando 1 conexão para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

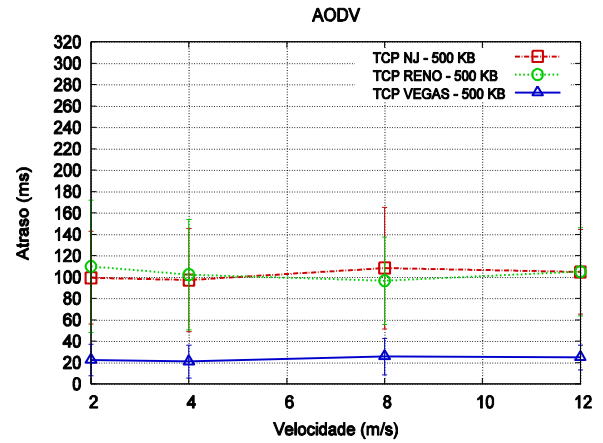
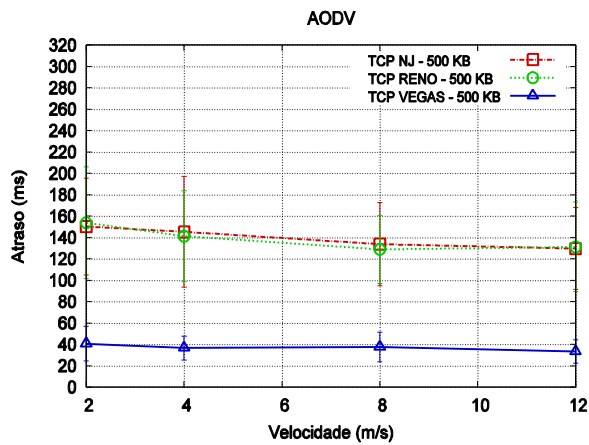


Figura A.2: Considerando 5 conexões simultâneas para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

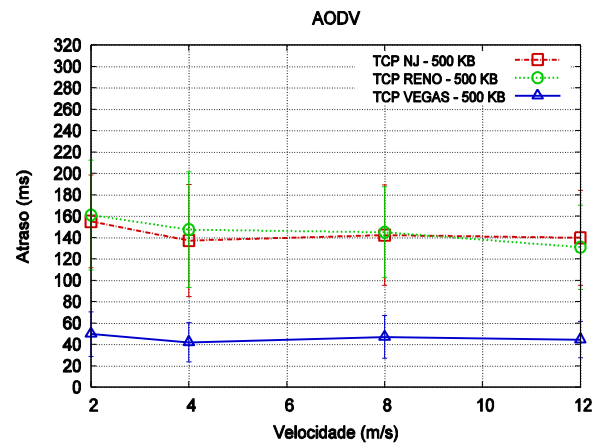
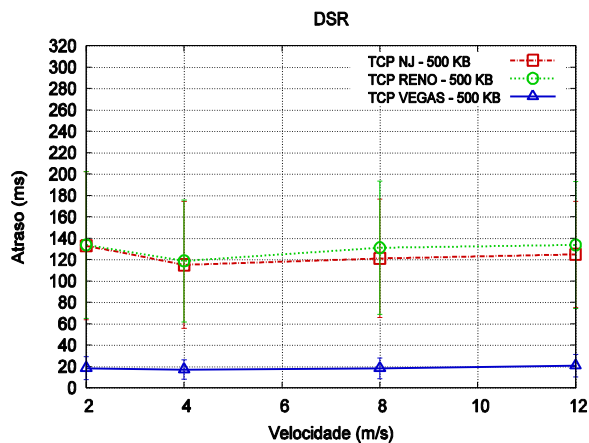


Figura A.3: Considerando 1 conexão para o DSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

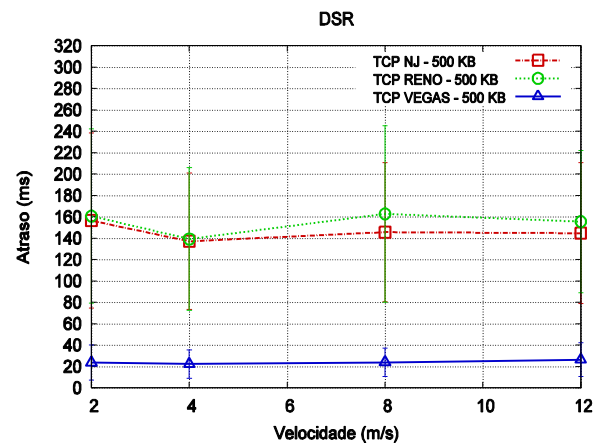


Figura A.4: Considerando 5 conexões simultâneas para o DSR.

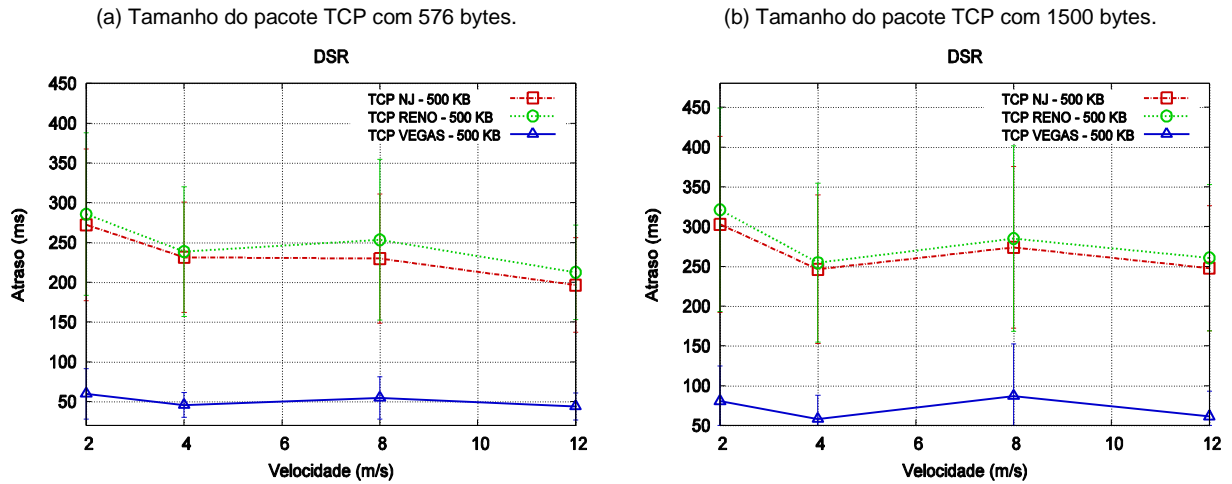


Figura A.5: Considerando 1 conexão para o OLSR.

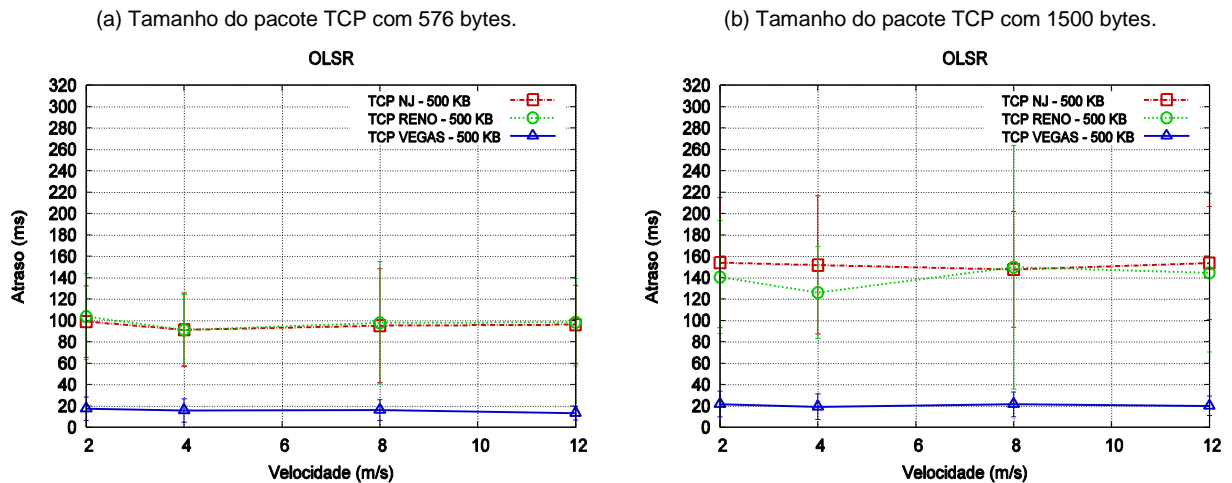
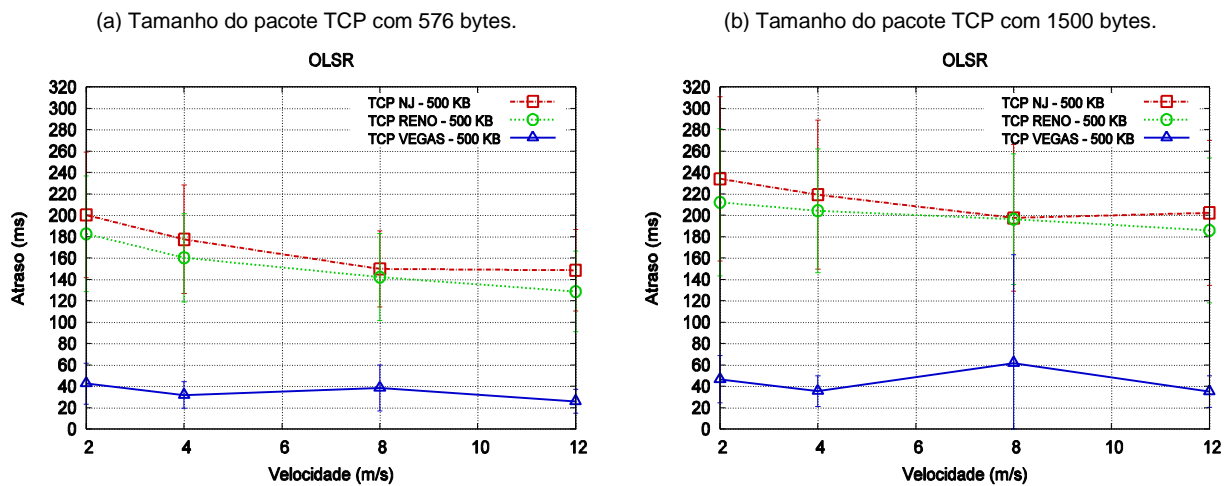


Figura A.6: Considerando 5 conexões simultâneas para o OLSR.



Na métrica de atraso pode-se observar que a implementação do TCP que obtém o melhor desempenho sobre os protocolos de roteamento das MANETS é o TCP Vegas, já o TCP New Jersey e Reno possuem desempenho semelhante. Todas as implementações do TCP têm melhor desempenho com o tamanho do pacote TCP definido em 576 bytes.

DESCARTE DE PACOTES

Figura A.7: Considerando 1 conexão para o AODV.

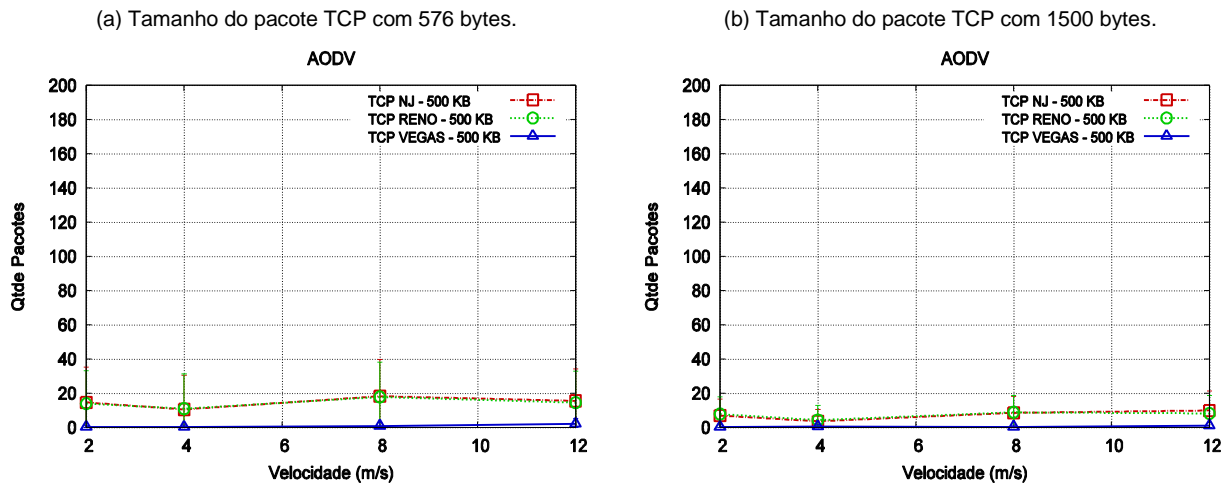


Figura A.8: Considerando 5 conexões simultâneas para o AODV.

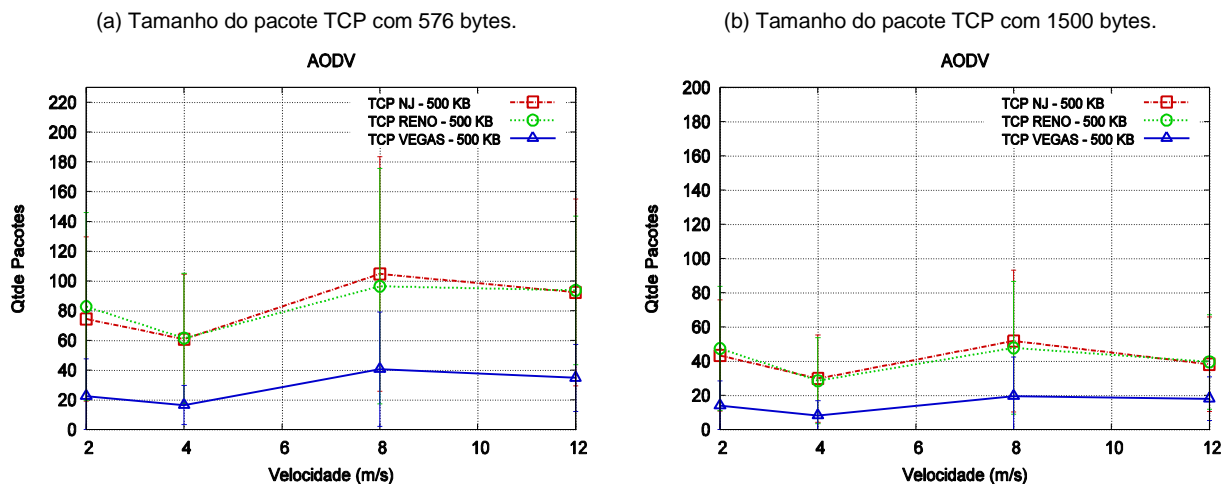
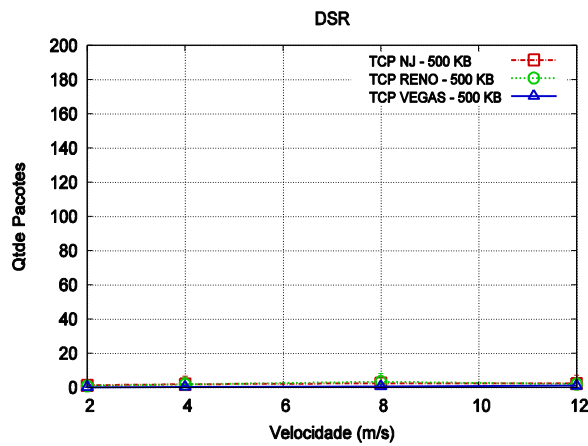
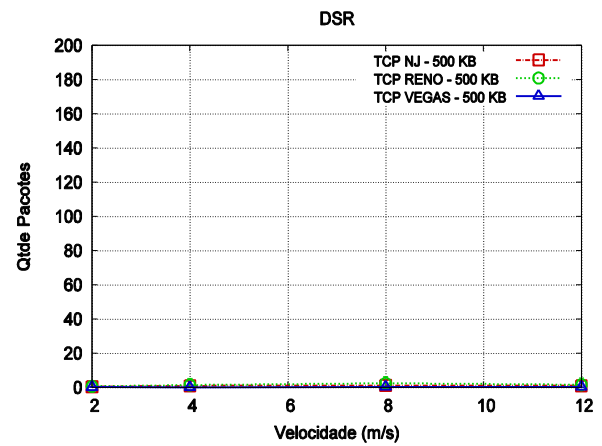


Figura A.9: Considerando 1 conexão para o DSR.

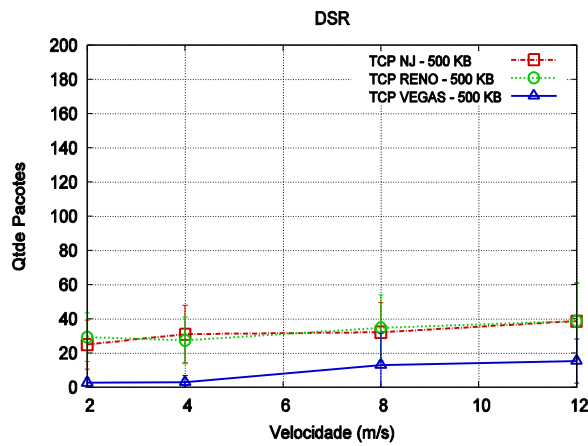
(a) Tamanho do pacote TCP com 576 bytes.



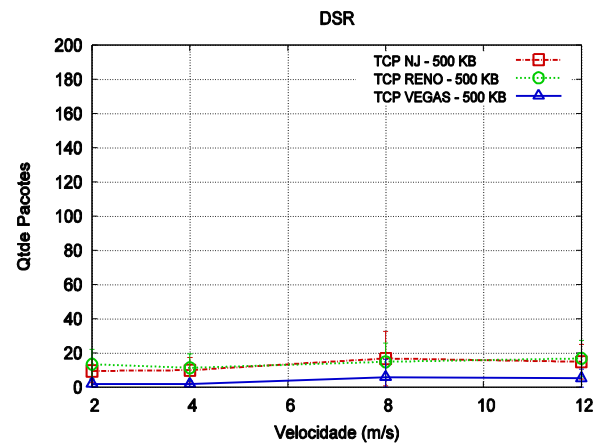
(b) Tamanho do pacote TCP com 1500 bytes.

**Figura A.10:** Considerando 5 conexões simultâneas para o DSR.

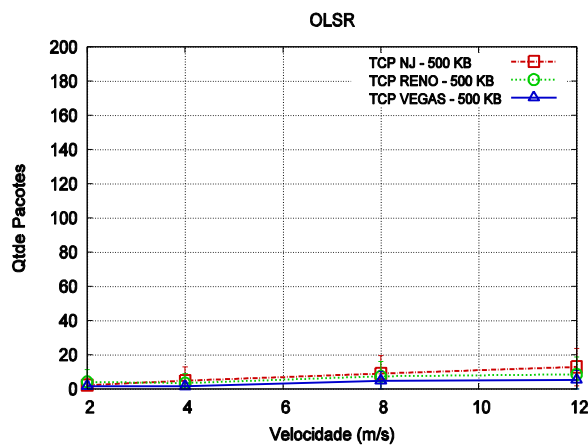
(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

**Figura A.11:** Considerando 1 conexão para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

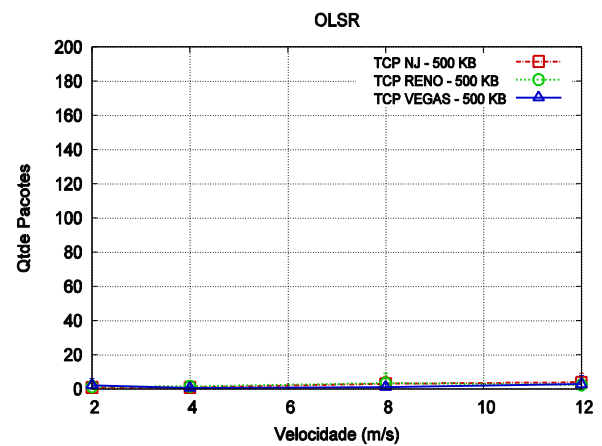
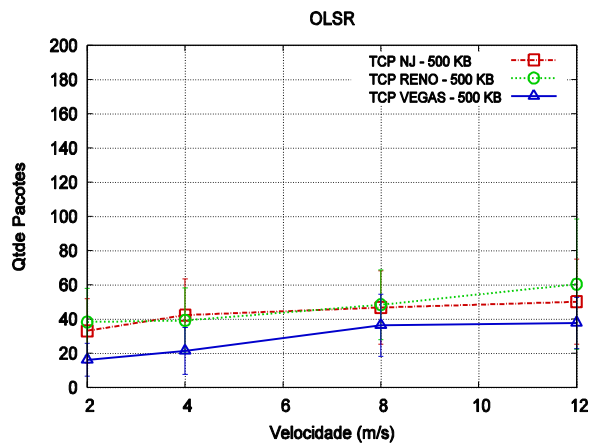
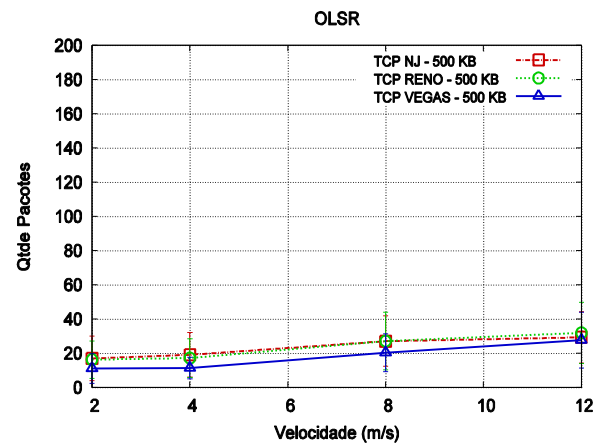


Figura A.12: Considerando 5 conexões simultâneas para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

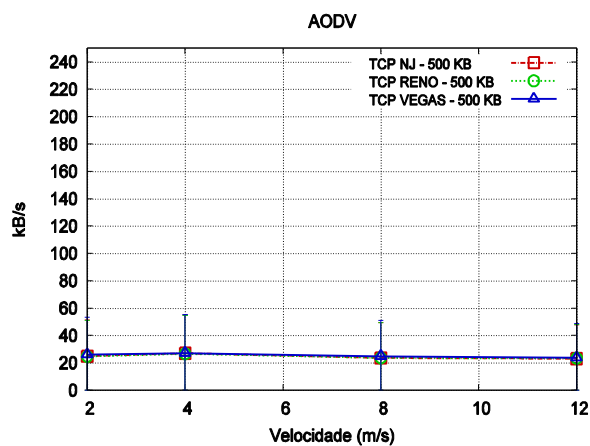


A métrica de descarte de pacotes demonstra que a implementação do TCP que têm melhor desempenho sobre os protocolos de roteamento das MANETS é o TCP Vegas, já o TCP New Jersey e Reno obtêm desempenho semelhantes. As implementações TCP utilizadas adquirem melhor desempenho com o tamanho do pacote TCP definido em 1500 bytes.

VAZÃO EFETIVA.

Figura A.13: Considerando 1 conexão para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

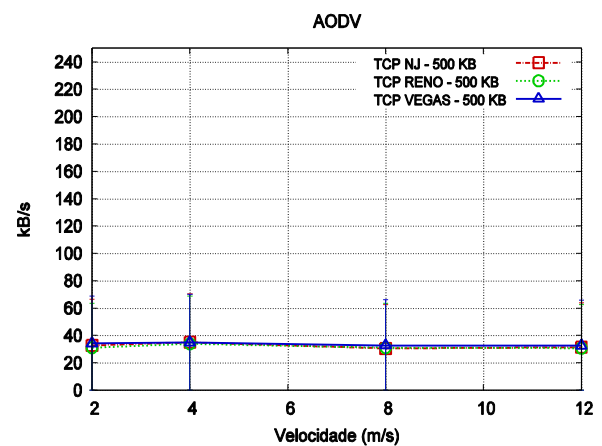
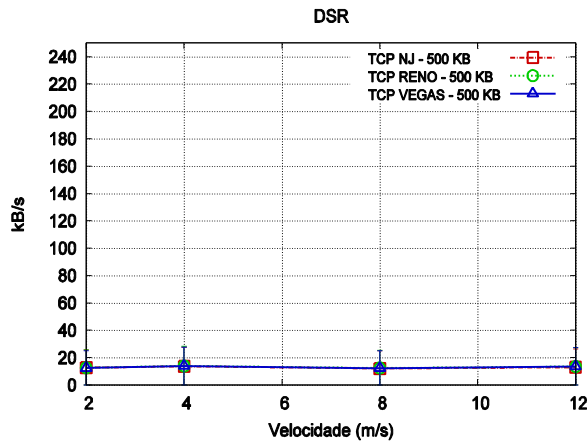


Figura A.14: Considerando 5 conexões simultâneas para o DSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

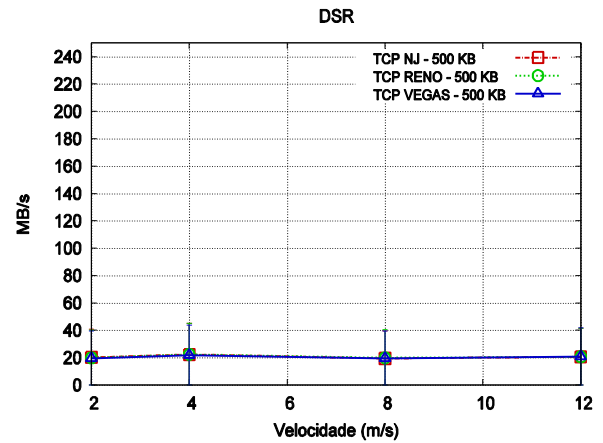
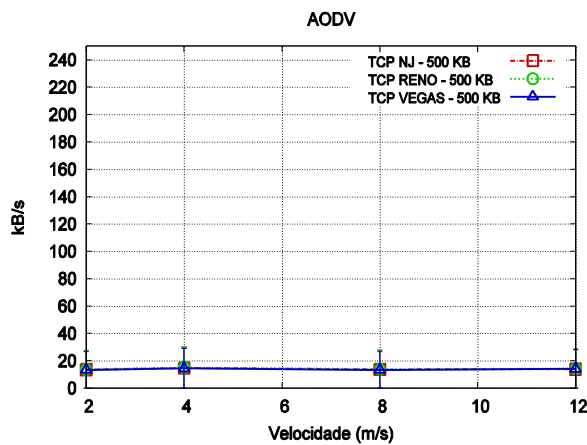


Figura A.15: Considerando 5 conexões simultâneas para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

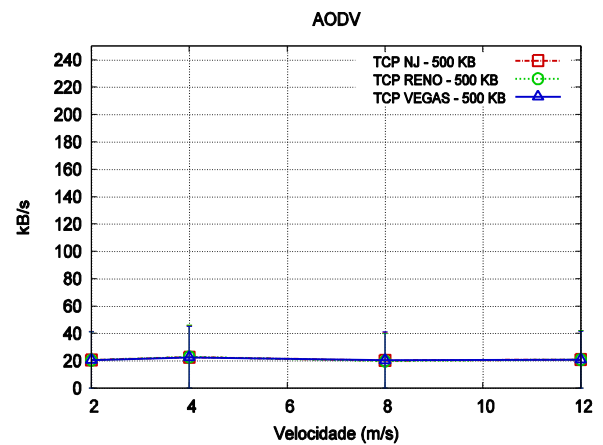
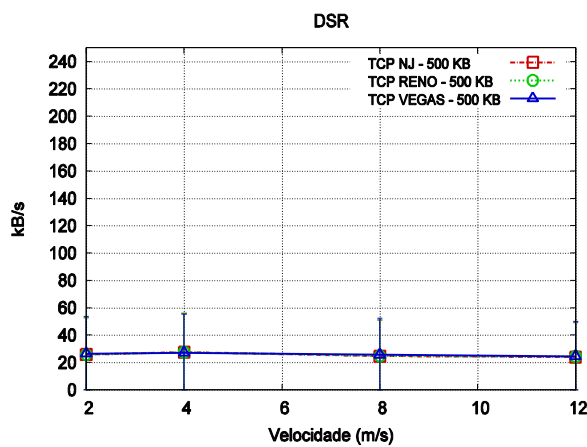


Figura A.16: Considerando 1 conexão para o DSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

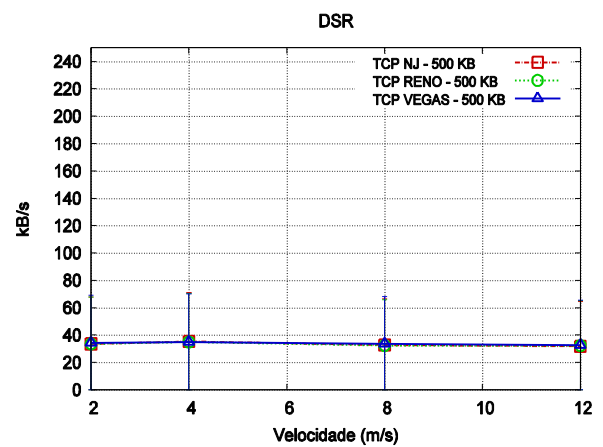
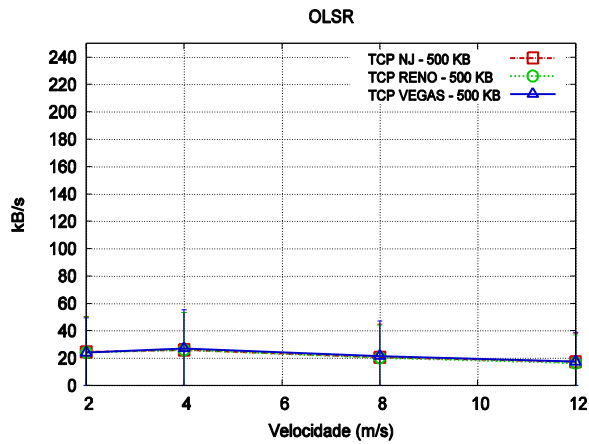


Figura A.17: Considerando 1 conexão para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

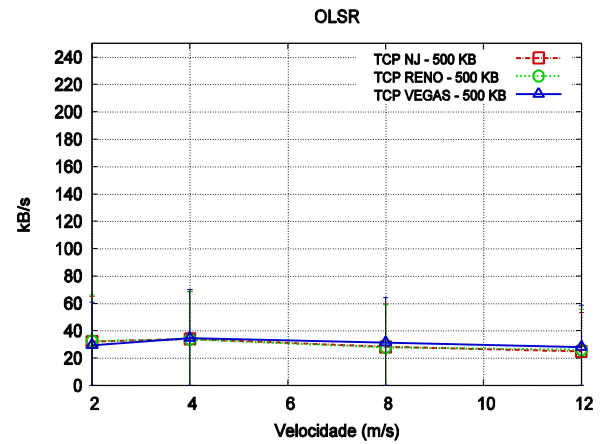
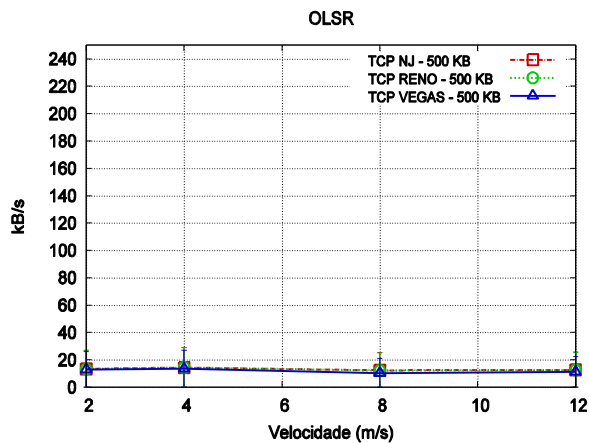
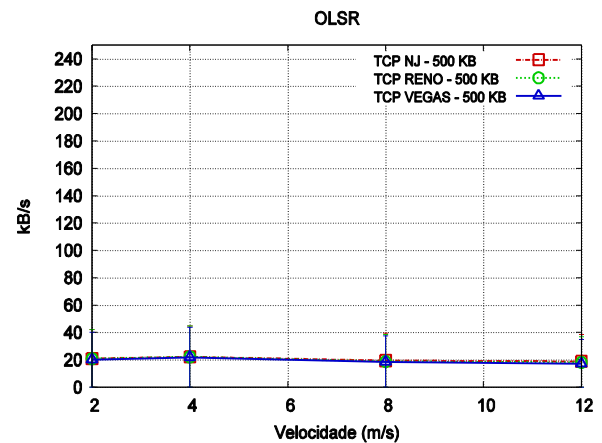


Figura A.18: Considerando 5 conexões simultâneas para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

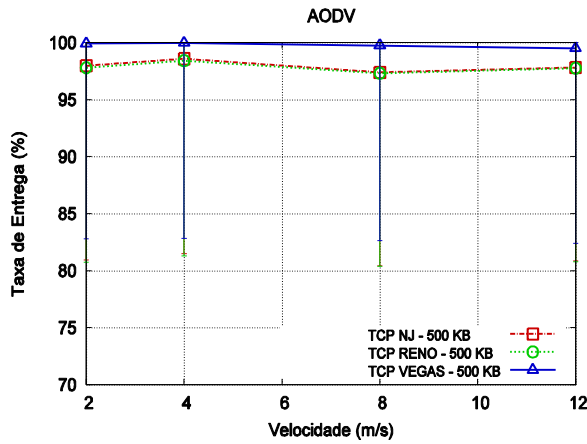


O desempenho das implementações do TCP na métrica de vazão efetiva é semelhante, tendo melhor desempenho com o tamanho do pacote TCP definido em 1500 bytes.

TAXA DE ENTREGA.

Figura A.19: Considerando 1 conexão para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

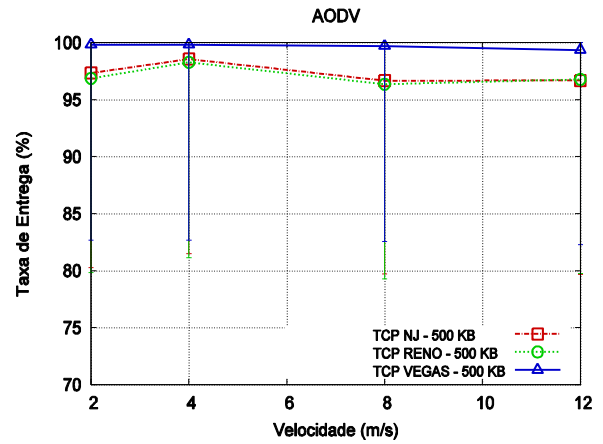
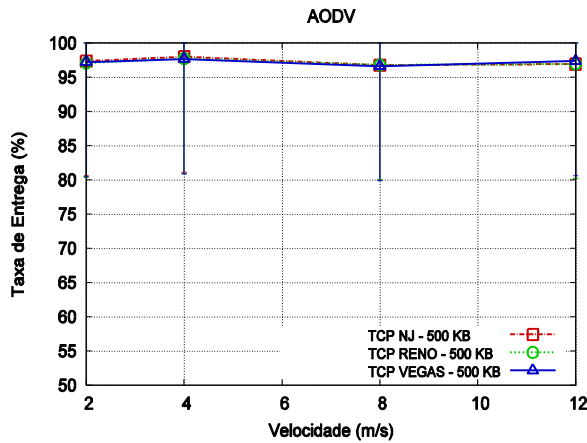


Figura A.20: Considerando 5 conexões simultâneas para o AODV.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

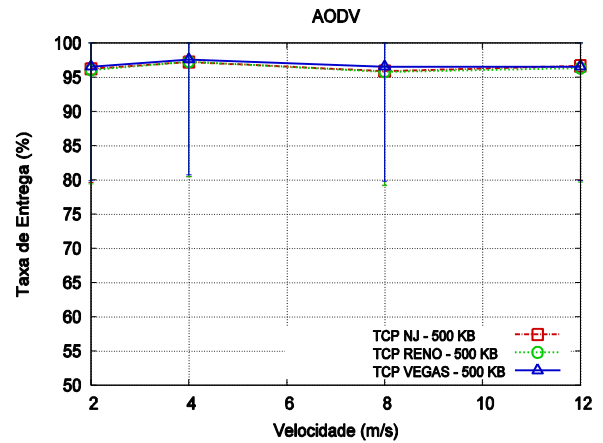
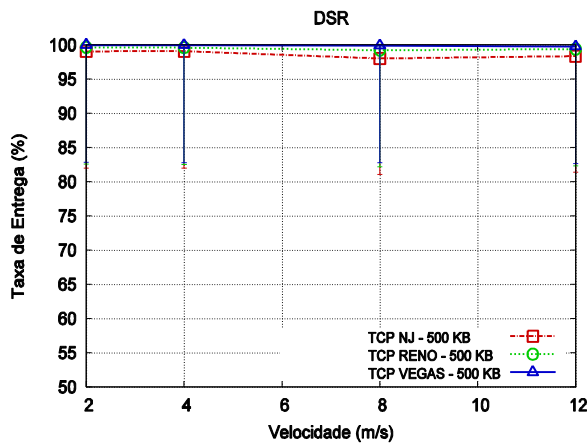


Figura A.21: Considerando 1 conexão para o DSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

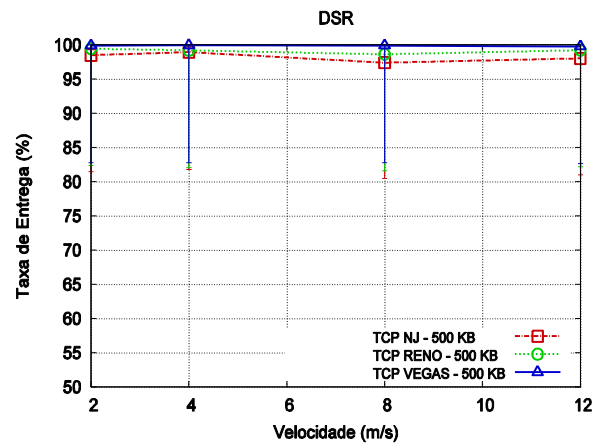


Figura A.22: Considerando 5 conexões simultâneas para o DSR.

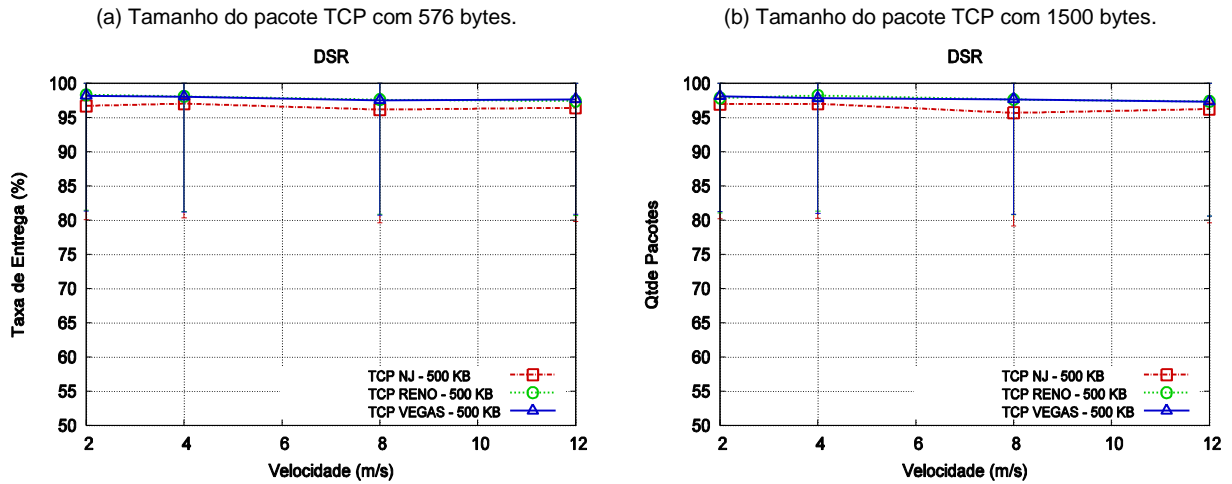


Figura A.23: Considerando 1 conexão para o OLSR.

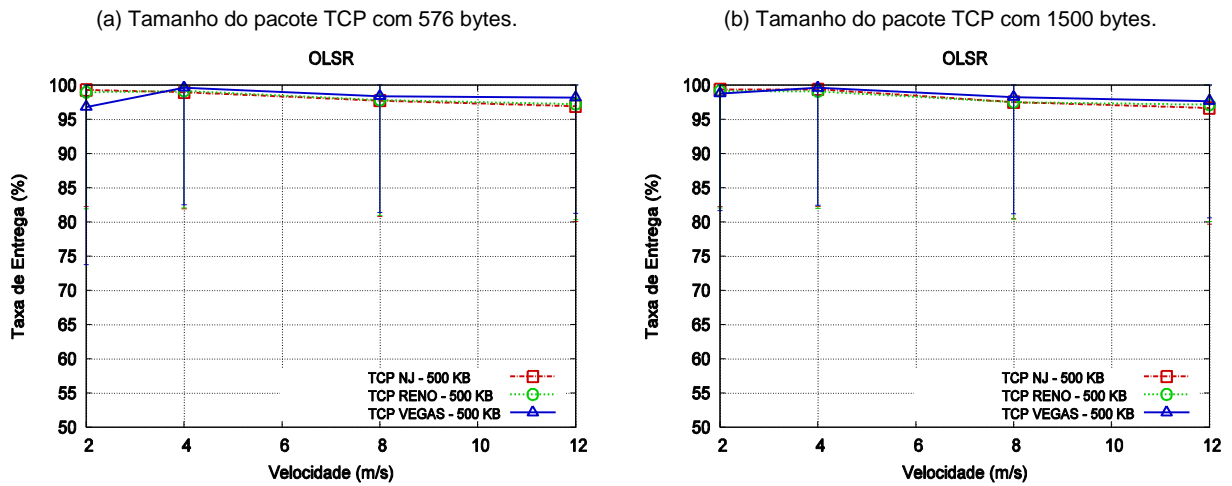
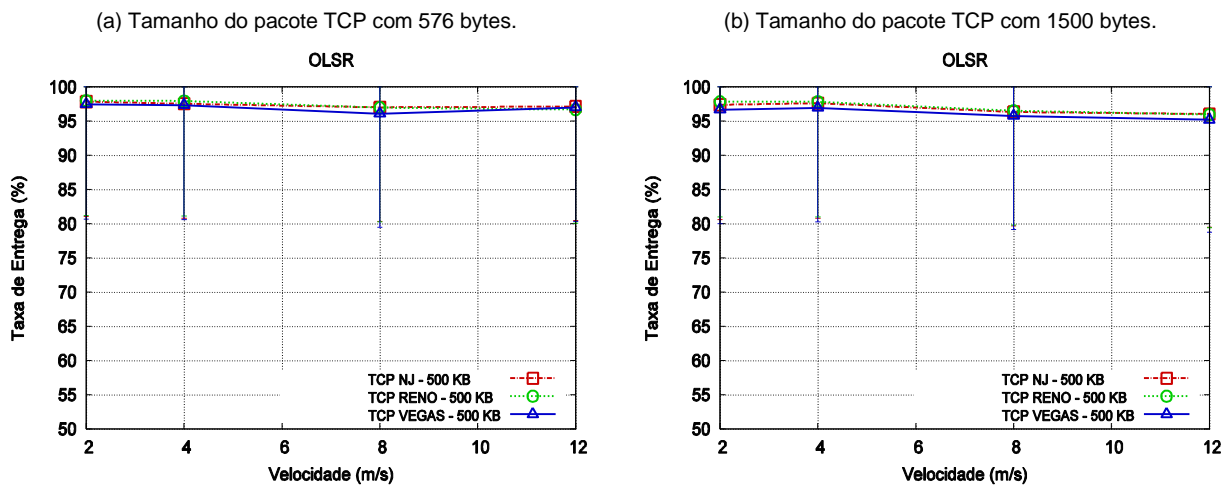


Figura A.24: Considerando 5 conexões simultâneas para o OLSR.



A métrica de vazão efetiva da rede, demonstra que com a utilização de somente uma conexão sendo realizada sobre a rede o desempenho das implementações é melhor que ao utilizar 5 conexões, já a utilização de 1500 bytes para o tamanho do pacote TCP não causa melhora significativa no desempenho das implementações.

TEMPO TOTAL DE ENVIO DE DADOS.

Figura A.25: Considerando 1 conexão para o AODV.

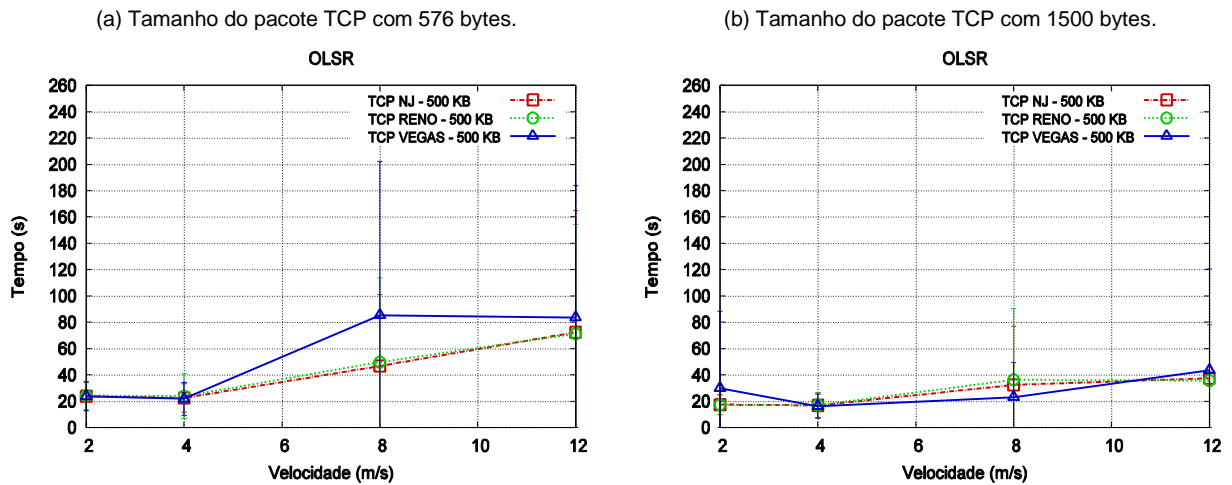


Figura A.26: Considerando 5 conexões simultâneas para o AODV.

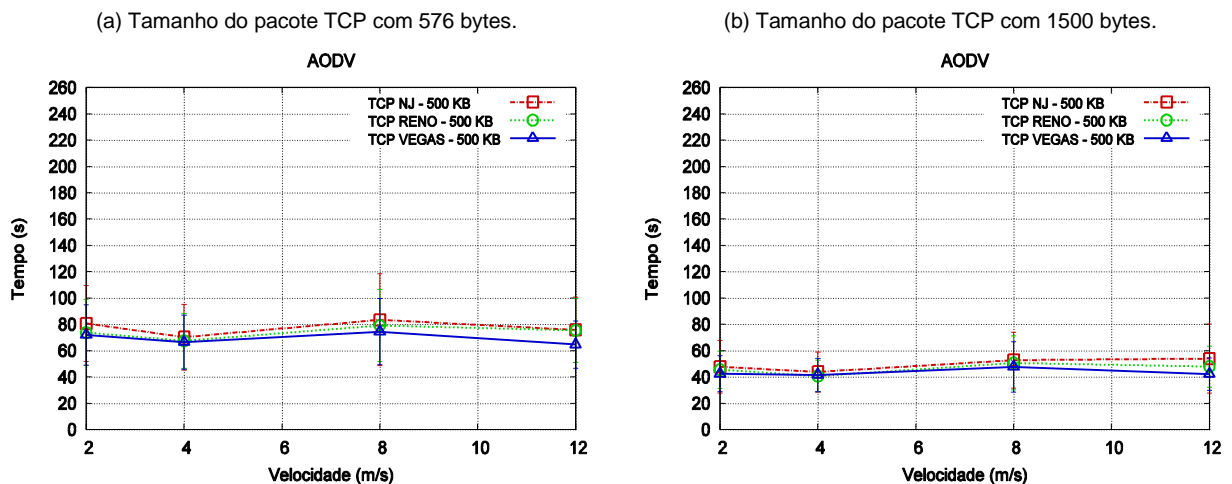
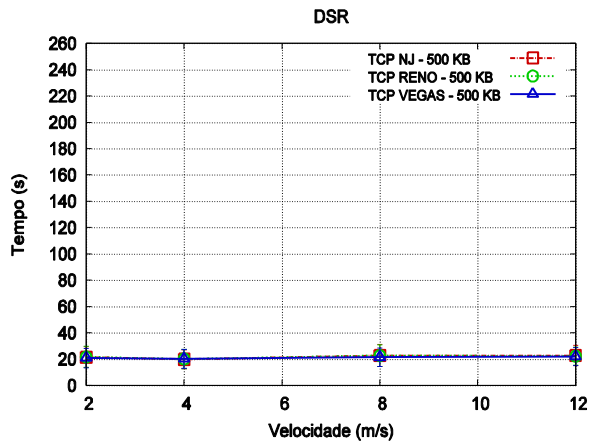


Figura A.27: Considerando 1 conexão para o DSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

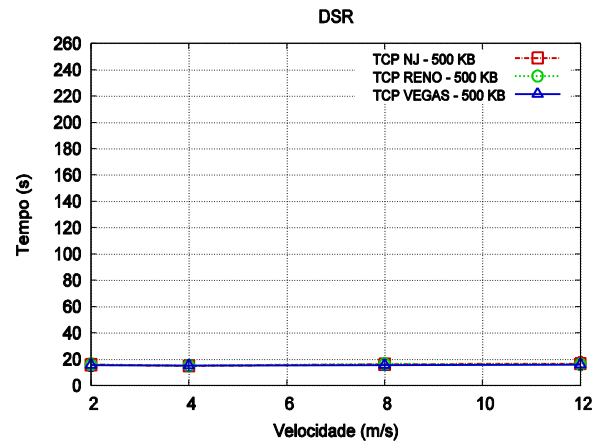
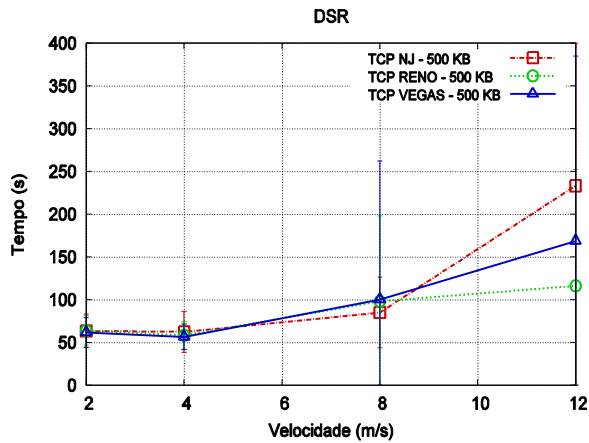


Figura A.28: Considerando 5 conexões simultâneas para o DSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

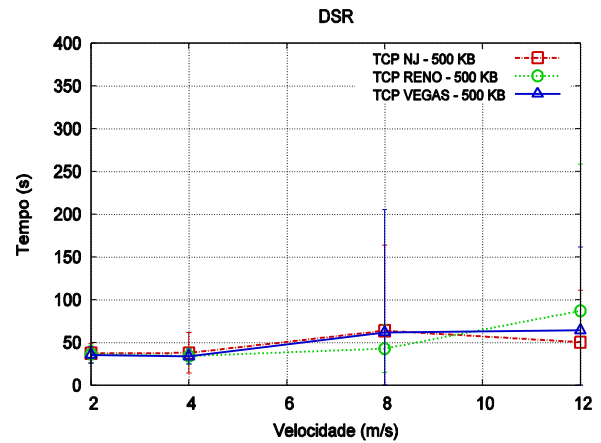
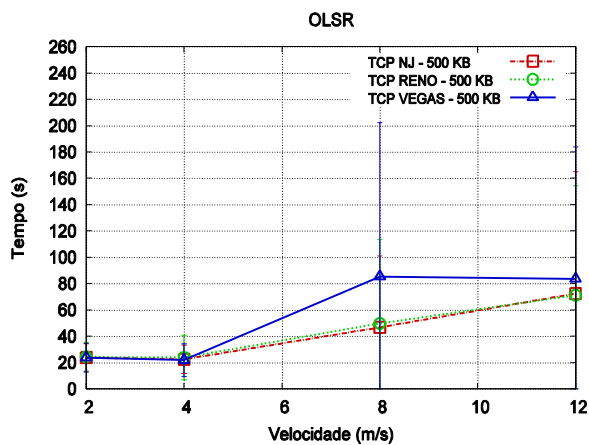


Figura A.29: Considerando 1 conexão para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.

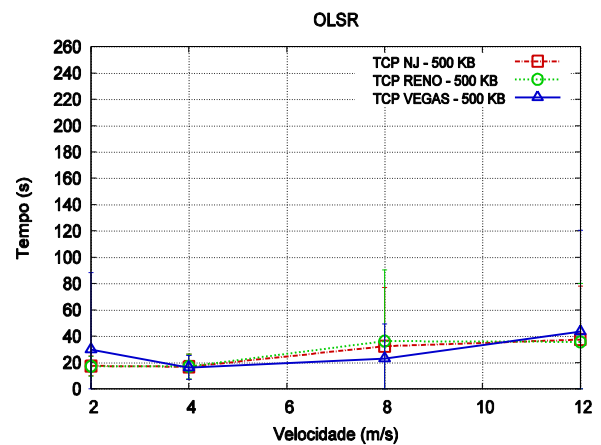
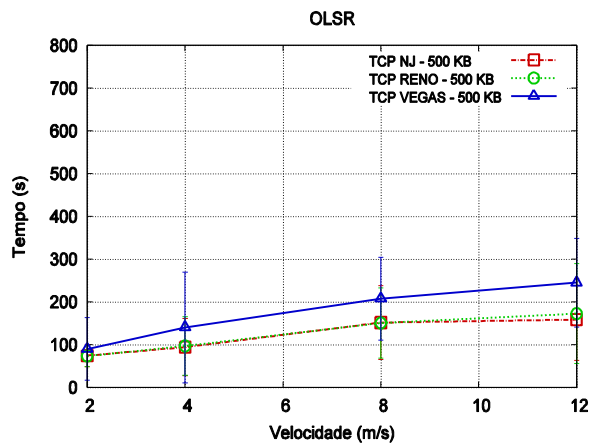
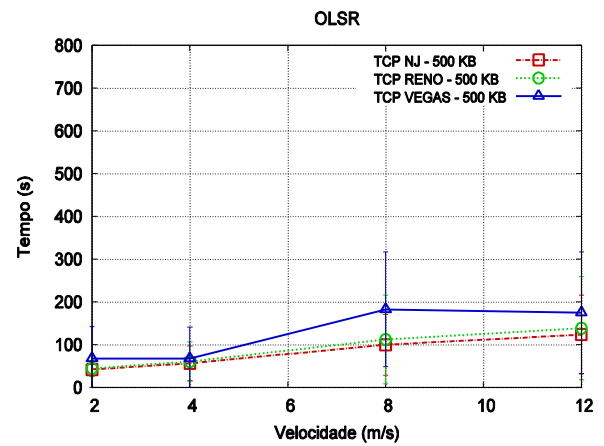


Figura A.30: Considerando 5 conexões simultâneas para o OLSR.

(a) Tamanho do pacote TCP com 576 bytes.



(b) Tamanho do pacote TCP com 1500 bytes.



O desempenho das implementações na métrica de tempo total de envio de dados, é melhor com o tamanho do pacote TCP definido em 1500 bytes, sendo que com o aumento do número de conexões realizadas pela rede, pior é o desempenho das implementações.