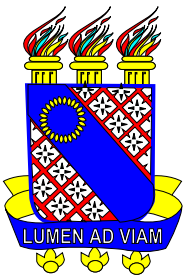

UNIVERSIDADE ESTADUAL DO CEARÁ

ANDRÉ JACKSON GOMES BESSA

UMA EXTENSÃO AO PROTOCOLO OLSR PARA ROTEAMENTO DE DADOS EM TEMPO REAL UTILIZANDO MÚLTIPLOS CAMINHOS

FORTALEZA, CEARÁ

2010

ANDRÉ JACKSON GOMES BESSA

UMA EXTENSÃO AO PROTOCOLO OLSR PARA ROTEAMENTO DE DADOS EM TEMPO REAL UTILIZANDO MÚLTIPLOS CAMINHOS

Dissertação apresentada no Mestrado em Ciências da Computação da Universidade Estadual do Ceará, como requisito parcial para obtenção do grau de Mestre em Redes de Computadores.

Orientador: André Ribeiro Cardoso

Co-Orientadores: Jorge Luiz Castro e Silva

FORTALEZA, CEARÁ

2010

A000z Bessa, André Jackson Gomes.
Uma extensão ao Protocolo OLSR para Roteamento de Dados em Tempo Real Utilizando Múltiplos Caminhos / André Jackson Gomes Bessa. – Fortaleza, 2010.
70p.;il.
Orientador: Prof. Dr. André Ribeiro Cardoso
Monografia (Mestrado em Ciências da Computação) - Universidade Estadual do Ceará, Centro de Ciências Científicas.
1. Ad Hoc 2. Tempo Real 3. RTP I. Universidade Estadual do Ceará, Centro de Ciências Científicas.

CDD:000.0

ANDRÉ JACKSON GOMES BESSA

UMA EXTENSÃO AO PROTOCOLO OLSR PARA ROTEAMENTO DE DADOS EM TEMPO REAL UTILIZANDO MÚLTIPLOS CAMINHOS

Dissertação apresentada no Mestrado em Ciências da Computação da Universidade Estadual do Ceará, como requisito parcial para obtenção do grau de Mestre em Redes de Computadores.

Aprovada em: __/__/____

BANCA EXAMINADORA

Prof. Dr. Jorge Luiz Castro e Silva
Universidade Estadual do Ceará - UECE
Co-orientador

Prof. Dr. Joaquim Celestino Júnior
Universidade Estadual do Ceará - UECE

Prof. Dr. Rommel Wladimir de Lima
Universidade Estadual do Rio Grande do Norte - UERN

Prof. Dr. Pedro Klecius Farias Cardoso
Instituto Federal do Ceará - IFCE

AGRADECIMENTOS

Primeiramente agradeço a Deus pela vida e por me permitir alcançar este tão esperado sonho. Obrigado pela saúde e pela família maravilhosa que me deste e por tantas pessoas sem as quais este trabalho não seria possível.

Agradeço ao meu orientador, o professor Dr. André Ribeiro Cardos, ao meu co-orientador, o professor Dr Jorge Luiz Castro e Silva e ao professor Dr Joaquim Celestino Júnior, pelas orientações e por acreditar na possibilidade da realização desta pesquisa. Sem seus esforços e ajuda, certamente esse trabalho não seria possível.

Aos membros da banca, o professo Dr. Rommel Wladimir de Lima e o professor Dr.Pedro Klecius Farias Cardoso que tão gentilmente se colocarão a disposição para avaliar esse trabalho. Muito obrigado a vocês.

Agradeço ao Mestrado Acadêmico em Ciências da Computação da Universidade Estadual do Ceará (UECE) pelo apoio ao desenvolvimento dessa pesquisa. Aos amigos do LARCES pela amizade e pelas ajudas nos momentos de dificuldade no decorrer desse período. Um agradecimento especial ao amigo e companheiro Sérgio Luiz, Davi di Cavalcanti, Sávio Moreira, Felipe Marciel e Rudy Matela, pelos bons momentos, pelas madrugadas de estudo e por seu apoio e por compartilhar seus ricos conhecimentos, que muito ajudaram na realização desse projeto.

Faço uma dedicação toda especial a minha esposa Márcia Angélica dos Santos Lima, pela paciência e compreensão, pois fomos privados de estarmos juntos em vários momentos especiais para que esse trabalho fosse realizado.

Aos meus pais, Maria do Socorro Gomes Bessa e Evaristo Bessa, pela ajuda, por acreditar nesse filho e por sempre estar presente quando precisei. As minhas queridas irmãs Ana Claudya e Auryleda Gomes Bessa, por tudo que fizeram por mim, sem dúvida sem vocês nunca chegaria até aqui. Enfim a todos os familiares por serem a melhor família que eu poderia ter.

"Um homem nunca sabe aquilo de que é capaz até que o tenta fazer"
Charles Dickens

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

As redes infraestruturadas oferecem um ambiente controlado de transmissão de dados. Isso tornou possível o uso de aplicações multimídia. Essas aplicações são utilizadas para consultar conteúdos armazenados em servidores ou para comunicação em tempo real. O *Real Time Transporte Protocol*, ou RTP, é bastante utilizado por essas aplicações. Ele funciona em conjunto com UDP e não possui mecanismos de controle de qualidade de transmissão. O responsável por prover esses dados para a aplicação é o *Real Time Control Protocol*, ou RTCP. As fontes e os destinos, ao longo da transmissão, trocam relatórios contendo informações como o número de bytes enviados e o atraso fim-a-fim observado. O uso do RTP em redes Ad Hoc móveis vem sendo estudado. As MANETs se caracterizam pela mobilidade de seus nós e, consequentemente, a constante modificação dos enlaces, tornando o roteamento uma tarefa complicada. Nesse ambiente, o uso do RTP para transmissão de dados armazenados em servidores vem obtendo algum êxito, porém a transmissão em tempo real ainda é um desafio. Este trabalho apresenta uma extensão ao protocolo de roteamento OLSR chamado de OLSR MultiCaminhos para Tempo Real (MCTR). Essa variação do OLSR é capaz de utilizar múltiplos caminhos para transmitir dados em tempo real. Ele faz uso de uma estratégia chamada *cross-layer*. Essa estratégia permite a mais de uma camada tratar os dados presentes na mesma região de um pacote, contrariando os princípios do padrão OSI. As informações utilizadas pelo OLSR MCTR estão presentes nos relatórios RTCP. A partir delas, o OLSR MCTR verifica quantos caminhos são necessários para se obter uma transmissão de boa qualidade em uma rede Ad Hoc. E ssa abordagem foi denominada roteamento orientado a conteúdo. Para validar a proposta, foram geradas simulações no *Network Simulator* 2 (NS-2). As simulações compararam o OLSR MCTR ao OLSR, ao AODV e ao AOMDV. Os resultados indicaram em quais condições o uso do OLSR MCTR é considerado viável.

Palavras-Chave: Ad Hoc, Tempo Real, RTP, OLSR, Roteamento

ABSTRACT

Infra-structured networks offer a controlled environment of data transmission. This makes possible the use of multimedia applications. These applications are used to consult contents stored at servers or for real time communication. Real Time Transport Protocol, or RTP, is much used by these applications. It functions as a whole with UDP and there are no communication quality control devices. The responsible one to provide theses data to the application is Real Time Control Protocol, or RTCP. The sources and the destinations, along transmission, change reports which have information like number of send bytes and end-to-end delay observed. The RTP use in Ad Hoc mobile networks has been studied. MANETs characterized by their nodes mobility and, consequently, the constant links change, making the delivery a complicated task. In this environment, the use of RTP to transmission of data stored at a server is getting some successful, although the transmission in real time is still a challenge. This work presents an extension to OLSR delivery protocol called OLSR Multipath for Real Time (MPRT). This variation of OLSR is able to use multiple ways to transmit data in real time. It uses a strategy called cross-layer. This strategy allows more than one layer treats the data present at the same region in a package going against the OSI pattern principles. Information used by OLSR MPRT are at the RTCP reports. From them, OLSR MPRT verifies how many ways are necessary to get a good quality transmission in an Ad Hoc network. This approach was called oriented to data delivery. To validate the proposition, simulations at Network Simulator 2 (NS- 2) were generated. The simulations compared OLSR MPRT to OLSR, to AODV and to AOMDV. The results indicated in which conditions the use of OLSR MPRT is considered practicable.

Keywords: Ad Hoc, Real Time, RTP, OLSR, Routing

1 INTRODUÇÃO

Neste capítulo será abordada a motivação e a relevância deste trabalho para as redes de computadores - seção 1.1, os seus objetivos e contribuições - seção 1.2, e a sua organização, seção 1.3.

1.1 Motivação

É cada vez mais comum o uso de aplicações capazes de prover serviços de comunicação de voz e vídeo em tempo real. Essas aplicações substituem, com vantagens, o uso de aparelhos telefônicos, pois permitem uma comunicação de boa qualidade utilizando a internet como meio de transmissão. Por meio delas, é possível realizar desde um simples telefonema até vídeo conferências.

Para esse tipo de transmissão ser possível é necessário o uso de um protocolo de transmissão apropriado. Esse protocolo deve ser capaz de transmitir dados de forma que eles alcancem o seu destino no menor tempo possível. Segundo a *International Telecommunication Union* (ITU), uma transmissão de áudio de boa qualidade, em tempo real, exige um atraso máximo de 150 ms entre a origem e o destino, sendo aceitável até 400 ms para alguns tipos de aplicações.

Hoje em dia, o protocolo mais utilizado para transmitir esse tipo de tráfego é o RTP (AL; SAADAWI; LEE, 2007) em conjunto com o RTCP (AL; SAADAWI; LEE, 2007). O *Real Time Transport Protocol*, ou RTP (SCHULZRINNE et al., 2003), e o *Real Time Control Protocol* foram criados juntos. O RTP é responsável pelo envio dos dados enquanto o RTCP monitora a qualidade da transmissão através da troca entre origem e destino dos chamados relatórios RTCP. Os relatórios RTCP são formados por uma série de informações (atraso, variação do atraso, quantidade de pacotes recebidos, etc.) utilizadas para regular a forma como a transmissão esta sendo executada pelo RTP. Em redes infraestruturadas o RTP apresenta um excelente funcionamento, porém em redes Ad Hoc móveis o seu desempenho ainda é abaixo do desejado.

As redes Ad Hoc móveis são redes não infraestruturadas de transmissões de dados, formadas por dispositivos móveis capazes de trocar informações entre si. Diferentemente das redes infraestruturadas, uma rede Ad Hoc não possui uma administração central. Nessas redes, cada um dos dispositivos funciona como um roteador com alcance limitado de transmissão. Devido ao constante deslocamento dos nós, a sua topologia muda muito; enlaces estão se formando e se rompendo o tempo todo. A velocidade de modificação da topologia das redes Ad

Hoc móveis torna complexo o roteamento do fluxo de dados. Assim, quanto maior a dificuldade na determinação de um caminho, maior o atraso e pior o desempenho das aplicações. Dessa forma, torna-se desafiador o uso dos protocolos RTP/RTCP, conforme concebidos, em redes Ad Hoc móveis. Atualmente eles são objetos de pesquisa na comunidade de redes (MAO et al., 2005).

Alturki avaliou o desempenho da transmissão de dados multimídia em redes Ad Hoc em seu trabalho (ALTURKI; MEHMOOD, 2008) e constatou, ao longo do experimento, a queda da vazão de dados em até 80% àmedida que novos nós eram adicionados a rede. A queda da vazão foi provocada por falhas de links ao longo da rede.

Vários trabalhos (AL; SAADAWI; LEE, 2007), (LIU; LIU, 2005), (MAO SHUNAN LIN, 2003), (LIN et al., 2004), (QIN; ZIMMERMANN, 2007) estão sendo lançados propondo soluções. Os melhores resultados estão sendo alcançados com o uso do RTP para transmissão de *streams* de áudio e vídeo sob demanda, baseados em arquivos armazenados em servidores. Porém, transmissão de conteúdo em tempo real se mostra um campo aberto a novas pesquisas (MAO SHUNAN LIN, 2003).

O trabalho de Mao, Bushmitch, Narayanan e Panwar (MAO et al., 2006a) avalia o uso de múltiplos caminhos para a transmissão de dados em redes Ad Hoc. Os seus resultados apontam para o aumenta da confiabilidade da transmissão. Simulações indicaram ganho de desempenho com o uso de dois caminhos para a transmissão de dados. Neste trabalho o autor demonstra um mecanismo de seleção da quantidade de caminhos.

Um protocolo de roteamento apropriado para trabalhar em conjunto com o RTP, transmitindo dados em tempo real, deve ser capaz de prover rapidamente o caminho conforme solicitado pela aplicação. Nessas condições, os protocolos de roteamento proativos levam vantagem sobre os reativos, pois já têm previamente calculado todos os nós a seu alcance, ao contrário dos reativos.

Este trabalho propõe uma extensão para o OLSR chamada OLSR MultiCaminhos para Tempo Real (MCTR). Esse novo protocolo funciona de forma similar ao OLSR, porém é capaz de prover múltiplos caminhos para aplicações. A quantidade de caminhos utilizados é definida em função dos dados presentes nos relatórios RTCP. O uso de dados de outros protocolos no OLSR MCTR se deve ao fato dele fazer uso de uma estratégia chamada de *cross layer*.

Essa técnica vem sendo utilizada em vários trabalhos, (CONTI; GREGORI; TURI, 2005), (LI et al., 2009),(CONTI; GREGORI; TURI, 2005),(VUTUKURU; BALAKRISHNAN; JAMIESON, 2009),(BOURAS; GKAMAS; KIOUMOURTZIS,),(BOURAS,) envolvendo redes ad hoc. Segundo o modelo OSI (DAY, 1995), os pacotes carregam vários dados onde cada um deles deve ser tratado por uma camada específica, não sendo possível duas camadas tratarem o mesmo dado. A técnica cross layer consiste em permitir que os dados dos pacotes sejam tratados por mais de uma camada e com isso otimizar o funcionamento dos protocolos. Fazendo uso dessa técnica, o OLSR MCTR utiliza os dados presentes nos relatórios RTCP para avaliar a qualidade do roteamento disponibilizado para a aplicação e, caso necessário, fazer uso de procedimentos adequados para melhorá-la.

O desempenho do OLSR MCTR foi comparado ao OLSR padrão, o AODV e o AOMDV. Dessa forma, poderemos não só comparar o desempenho do OLSR MCTR com o original, como também é possível verificar o comportamento em comparação com protocolos reativos.

1.2 Objetivos e Contribuições

O objetivo desse trabalho é desenvolver um protocolo de roteamento para redes Ad Hoc capas de suportar transmissões de dados em tempo real. Esse protocolo se chama OLSR MCTR, ele foi desenvolvido a partir do OLSR. Diferentemente do OLSR padrão o OLSR MCTR é capaz de transmitir o fluxo de dados por vários caminhos distintos. A quantidade de caminhos escolhidos é determinada pelas informações obtidas nos relatórios RTCP. Essa abordagem foi denominada roteamento orientado a conteúdo. O trabalho proposto utiliza o simulador de rede NS-2.

Neste trabalho será apresentada a especificação do OLSR MCTR e a comparação do seu desempenho com o do OLSR, do AODV e do AOMDV. Os resultados foram avaliados em função do atraso fim-a-fim e da percentagem de pacotes efetivamente entregues. As principais contribuições deste trabalho são:

1. modificação do OLSR para suportar múltiplos caminhos;
2. proposta de uma abordagem *cross-layer* capaz de capturar dados do protocolo RTCP para utilizá-los na melhoria do processo de roteamento;
3. definição e implementação de um algoritmo capaz de relacionar o atraso fim-a-fim com a quantidade de caminhos utilizados no envio do fluxo de dados;
4. definição do conceito roteamento baseado em conteúdo.

Ao longo do desenvolvimento do trabalho uma contribuição colateral foi acrescentada: a implementação do RTP/RTCP presente no Network Simulator 2 *Network Simulator 2* (KUBINIDZE; GANCHEV; O'DROMA, 2004), utilizado no desenvolvimento do trabalho, apresenta uma falha no mapeamento das configurações dos protocolos RTCP em TCL. Essa falha não permitia a correta execução do protocolo em ambiente Ad Hoc. Esse problema foi detectado e resolvido para ser possível executar as simulações.

1.3 Organização do Trabalho

Neste Capítulo, apresenta-se uma seção introdutória expondo o contexto, a motivação do trabalho e os seus objetivos. O Capítulo 2 constitui uma revisão teórica dos tópicos:

1. Redes Ad Hoc Móveis, conceito, técnicas de roteamento e classificação dos protocolos;

2. Roteamento com múltiplos caminhos em Redes Ad Hoc Móveis, conceitos e desafios;
3. OLSR, formação do conjunto MPR, controle de topologia e cálculo da tabela de rotas;
4. AODV, apresentação e modo de funcionamento;
5. AOMDV, apresentação e modo de funcionamento;
6. RTP/RTCP;

O capítulo 3 descreve o funcionamento e a implementação do OLSR MCTR. O capítulo 4 descreve o simulador utilizado para avaliar o desempenho do OLSR MCTR em comparação com o OLSR, AODV e AOMDV, os cenários utilizados e as métricas adotadas. O capítulo 5 expõe os dados obtidos pelas simulações e os analisa determinando em quais condições o OLSR MCTR pode ser utilizado com ganhos. Finalmente, a revisão do trabalho, suas contribuições, os resultados alcançados e as propostas para trabalhos futuros são descritos no capítulo 6.

2 REVISÃO TÉCNICA

Neste capítulo será realizada a revisão dos principais assuntos discutidos nesse trabalho. A seção 2.1 apresenta uma breve definição de redes móveis Ad Hoc e de algumas de suas características. A seção 2.1.3 revisa o funcionamento do OLSR. A seção 2.1.4 revisa o funcionamento do AODV. A seção 2.1.5 revisa o funcionamento do AOMDV. A seção 2.2 revisa o funcionamento dos protocolos RTP e RTCP.

2.1 Redes Ad Hoc Móveis

As redes móveis Ad Hoc, ou *Mobile Ad Hoc Network*(MANET), se caracterizam por rápidas mudanças na configuração de sua topologia, pois os nós em uma MANET podem se conectar dinamicamente e se movimentar livremente.

Essa característica, mais o fato da transmissão de dados acontecer em meio aberto, torna os enlaces sujeitos a interferências e quedas. Por isso, o roteamento dos fluxos de dados nessas redes é uma tarefa complexa.

2.1.1 Roteamento em MANET

2.1.1.1 Técnicas Utilizadas para Roteamento

Devido à constante alteração de topologia, protocolos de roteamento, em redes MANET, tendem a fazer uso de técnicas dinâmicas no seu desenvolvimento. As principais, segundo SARKAR et al. (2007), são estado de enlace e vetor distância. No roteamento vetor distância, cada roteador mantém uma tabela de encaminhamento na qual armazena as informações de distância para todos os seus vizinhos. Um roteador troca informações com os seus vizinhos periodicamente para manter a sua tabela de encaminhamento sempre atualizada. A distância pode ser calculada com base em métricas, tais como o número de saltos, ou o atraso.

Se existem vários caminhos, o mais rápido será selecionado. A principal desvantagem dessa técnica é a lenta convergência, ocasionando o problema de ”contagem ao infinito”, ou seja, alguns roteadores continuamente aumentando a contagem de hops existentes na rede. Na técnica estado de enlace, cada nó periodicamente notifica o status atual de seus enlaces para todos os roteadores na rede. Quando ocorrer alguma mudança do estado de um enlace, serão enviadas notificações para todos os nós da rede. Essa técnica é chamada de inundação. Depois

de receber as notificações, todos os roteadores recalculam as suas rotas, de acordo com as novas informações de topologia. Desta forma, um roteador tem, pelo menos, uma imagem parcial de toda a rede. A distância entre dois nós pode ser calculada a partir de diferentes métricas, tais como o número de saltos, velocidade do enlace e perda de pacotes.

2.1.1.2 Classificação dos Algoritmos de Roteamento

Um dos mais populares métodos para classificar protocolos de roteamento se baseia em como a informação é adquirida e mantida pelos nós móveis. Dessa forma os protocolos são divididos em proativos, reativos e híbridos. Utilizando um protocolo de roteamento proativo, os nós em uma rede Ad Hoc continuamente enviam informações a respeito de suas rotas aos nós vizinhos. Dessa forma a topologia se mantem atualizada em todos os participantes da rede. Utilizando um protocolo de roteamento reativo, as rotas são procuradas apenas quando necessário. O procedimento de busca é encerrado quando um caminho é encontrado ou quando não é possível determinar uma rota após todas as tentativas. O uso de protocolos reativos previne interferências ocasionadas pelo mecanismo de inundação presente nos protocolos proativos. Entretanto, os pacotes podem sofrer com o atraso de transmissão devido à demora no procedimento de busca. Os protocolos híbridos utilizam estratégias comuns aos protocolos proativos e aos reativos. Dessa forma podem unir a eficiência no consumo de recursos, existente nos reativos, com a velocidade de resposta quanto à determinação de uma rota, existente nos proativos.

2.1.1.3 Roteamento com Múltiplos Caminhos

Devido à mobilidade dos seus nós, uma MANET é relativamente vulnerável à quebra de seus enlaces. Por esse motivo, devemos considerar o uso de múltiplos caminhos para a transmissão dessa forma tornando a comunicação mais confiável, pois caso a origem e o destino detectem a perda de uma rota, basta ignora-la e fazer uso da segunda via, sem haver a necessidade de iniciar um novo processo de descoberta de novas rotas. O comportamento de um protocolo de roteamento capaz de processar múltiplos caminhos consiste em descobrir as possíveis rotas, avaliar sua validade e alocar o tráfego nas rotas disponíveis. Os enlaces são divididos em disjuntos por enlace ou disjuntos por nós. Enlaces disjuntos por enlace ocorrem quando existem múltiplos caminhos compartilhando nós intermediários. Enlaces disjuntos por nós ocorrem quanto todos os caminhos passam por nós distintos. Além da capacidade de recuperação da rede ser aumentada com o uso de múltiplos caminhos, vários outros aspectos do roteamento em redes Ad Hoc são afetados. São eles:

1. performance;
2. balanceamento de carga;
3. segurança.

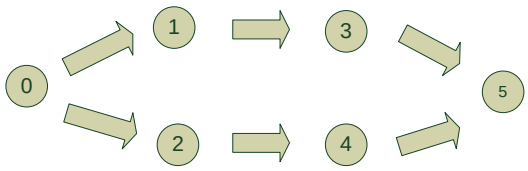

Figura 2.1: Caminho disjunto por nó

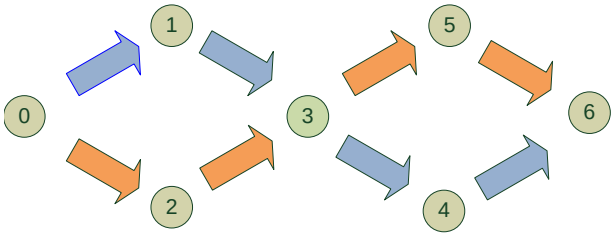

Figura 2.2: Caminho disjunto por enlace

A performance é aumentada, pois com a existência de mais de n caminhos é possível enviar, ao mesmo tempo, n pacotes. Assim, a capacidade de transmissão é ampliada. Caso o melhor caminho esteja sendo muito utilizado é possível diminuir a sua carga passando parte do fluxo por outro caminho, dessa forma distribuindo a carga existente na rede. Um pouco de segurança é adicionada à rede, pois mesmo que um dos caminhos esteja tendo seus pacotes capturados, apenas parte da comunicação estará sendo interceptada, não permitindo o acesso total ao conteúdo transmitido. O uso de múltiplos caminhos traz vantagens a uma rede ad hoc, através do aumento da taxa de dados, otimizando o consumo de energia, a robustez da rede, etc. No entanto, também existem alguns custos juntamente com os benefícios. O consumo de memória e a necessidade de um maior processamento, principalmente em redes densas, devem ser tratados com cuidado. Outra questão importante são as interferências nas transmissões. Dependendo do quão independente sejam os caminhos, a transmissão enviada por um nó pode causar ruídos na transmissão realizada por outro, limitando assim a taxa de transferência possível. Os caminhos, portanto, devem ser tão independentes quanto possível. Tudo isso torna o roteamento por múltiplos caminhos mais complexo em comparação com o roteamento convencional.

2.1.2 Protocolos de Roteamento para MANET

Existem vários protocolos de roteamento para as MANETs. Nesse trabalho estudaremos 3 protocolos. Os dois primeiros são o OLSR e o AODV, protocolos de roteamento que fazem uso de caminho simples. Por último estudaremos o AOMDV, protocolo de roteamento capaz de fazer uso de múltiplos caminhos.

2.1.3 OLSR

O *Optimized Link State Routing*(OLSR) (CLAUSEN; JACQUET, 2003) é um protocolo de roteamento proativo do tipo estado de enlace. Desta forma, periodicamente os nós inundam a rede com o estado de seus enlaces. Dessa forma todos os nós da rede podem construir uma representação da estrutura da rede.

A RFC 3626 especifica o protocolo OLSR. Ela detalha todos os componentes para o funcionamento do protocolo. A seguir serão descritos os componentes essenciais para o entendimento deste projeto:

- o cálculo do MPR;
- o formato dos pacotes;
- o cálculo da tabela de roteamento;
- a métrica de roteamento etx.

O que diferencia o OLSR de outros protocolos proativos é que ele utiliza os chamados Multipoint Relays (MPR), que servem para diminuir o número de mensagens de controle na rede. Quando um protocolo de roteamento de estado de enlace recebe pacotes de controle contendo informações sobre atualizações dos estados de enlace, ele retransmite essas informações para todos os seus vizinhos, esse mecanismo é denominado inundação. Dessa maneira cada nó pode receber o mesmo pacote diversas vezes dos seus vizinhos. O objetivo dos MPR é resolver esse problema limitando os nós responsáveis por fazer a inundação.

Assim, como cada nó possui seus MPR designados, a comunicação entre quaisquer nós da rede é feita pelos MPR dos nós intermediários. Essa otimização faz com que esse protocolo sobrecarregue menos a rede com tráfego de controle, principalmente para redes grandes e de alta densidade, sendo por outro lado melhor aplicável àquelas redes com pouca mobilidade, onde a topologia permaneça constante uma vez que não é gerado nenhum controle adicional de tráfego.

2.1.3.1 Formação do Conjunto MPR

Os nós MPR são responsáveis por informar o estado dos links da rede. Os nós que foram selecionados como MPR por algum nó vizinho anunciam essa informação periodicamente em suas mensagens de controle. O protocolo utiliza os MPR para facilitar a inundação das mensagens de controle de rede.

Para calcular o conjunto de nós MPR, cada nó seleciona, de maneira independente, entre seus vizinhos, com os quais possui um enlace simétrico. O conjunto de nós MPR será formado por nós capazes de atingir todos os vizinhos a dois saltos.

Para prover rotas para nós distantes a mais de 2 saltos, cada nodo mantêm informações sobre a topologia da rede. Essa informação é adquirida através de mensagens de controle de topologia (*Topology Control* - TC).

Os nós que foram selecionados como MPR por outros nós periodicamente geram mensagens TC, anunciando a lista de todos os nós seletores (MPR). Mensagens TC são disseminadas por *flooding* em toda a rede pelos MPR. Um campo de número de sequência de mensagem (SN) é usado para evitar o processamento duplicado de mensagens. Este campo é gerado como uma sequência de números inteiros, incrementada a cada mensagem gerada.

A escolha dos MPR por um nó se dá de maneira independente dos outros nós da rede. São utilizados apenas enlaces simétricos, que são anunciados aos demais nós através do campo tipo de vizinho presente na mensagem *HELLO*.

O algoritmo para a escolha dos MPR é executado para cada interface de rede física, sendo os MPR de um nó a união dos MPR de todas as interfaces. Toda vez que for detectada a entrada ou saída de um nó a até dois saltos de distância, o algoritmo de escolha dos MPR deve ser executado novamente.

O algoritmo para o cálculo do MPR é heurístico. Para realizá-lo, define-se:

- N = Conjunto dos vizinhos por um salto;
- N2 = Conjunto dos vizinhos por dois saltos, excluindo-se os membros de N e o próprio nó de origem;
- MPR = Conjunto de vizinhos selecionados como MPR;
- D(x) = Número de vizinhos com enlace simétricos de x excluindo-se todos os membros de N e o nó de origem, sendo x um membro de N;
- A(x) = Número de vizinhos de x em N2, sendo x um membro de N.

Resumindo, seu funcionamento é dado pelos seguintes passos:

1. Preencher os conjuntos N e N2 e deixar vazio o conjunto MPR;
2. Calcular D(x) para todo x que pertença a N;
3. Selecionar em N todos os nós que sejam os únicos a atingir algum membro em N2 e transferi-los para o conjunto MPR;
4. Excluir de N2 todos os nós vizinhos dos nós em MPR;
5. Calcular a alcançabilidade, que representa o número de nós em N2 que cada nó de N alcança;
6. Escolher o nó em N que possui a maior alcançabilidade entre os vizinhos em N2. Em caso de empate, escolher o que possui o maior D(x). Transferir o membro de N escolhido para MPR;

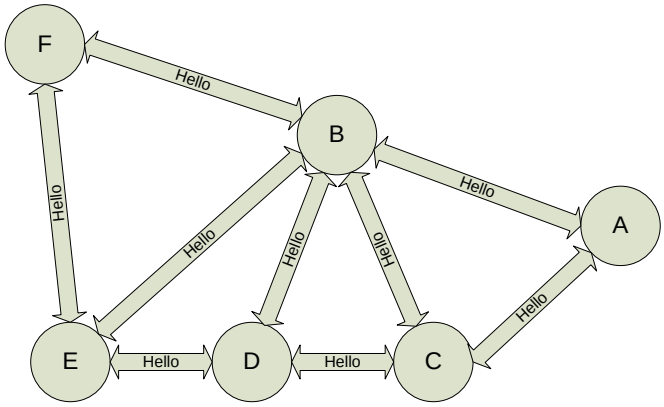

Figura 2.3: Os nós trocam mensagens *Hello*

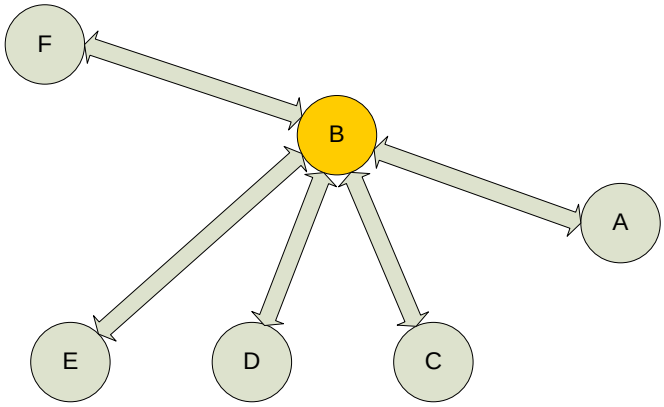

Figura 2.4: Nó B se tornou um MPR do nó A

7. Excluir de N2 todos os nós vizinhos do novo membro em MPR;
8. Se N2 não estiver vazio, voltar ao passo 5;
9. Se N2 estiver vazio, fim do algoritmo.

Assim, no caso de redes com mudanças de topologia constantes, como por exemplo, causadas por maior mobilidade, é recomendado que se tenha um conjunto maior de MPR redundantes para evitar o recálculo desse algoritmo.

2.1.3.2 Formato dos Pacotes

O OLSR tem todos os seus pacotes de controle enviados sobre UDP (*User Datagram Protocol*), utilizando a porta de número 698, concedida pela IANA. O formato básico dos pacotes do OLSR, omitindo os cabeçalhos IP e UDP, é descrito na Figura 2.5.

O campo *Packet Lenght* representa o tamanho do pacote todo enquanto o campo *Packet Sequence Number* é um número que deve ser incrementado toda vez que um pacote é transmi-

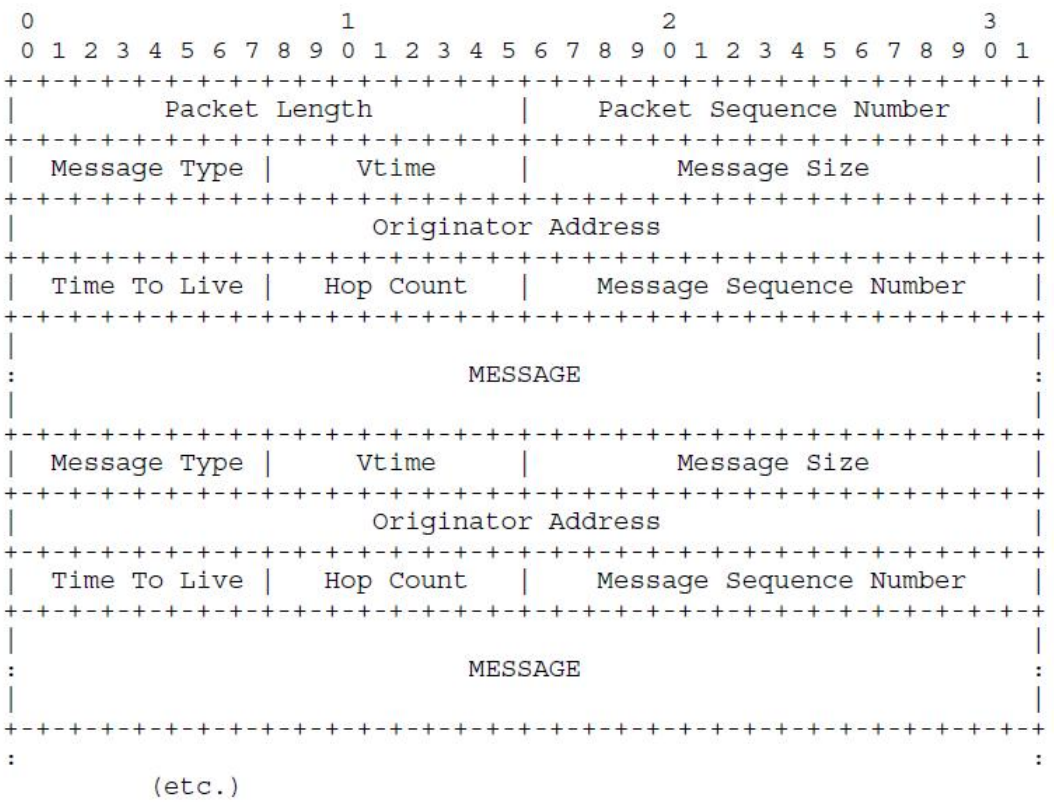

Figura 2.5: Formato de um pacote OLSR

tido, sendo utilizado apenas para a detecção de perda de pacotes. O campo *Message Type* representa o tipo de mensagem que é transmitida, como por exemplo, mensagens *HELLO* e mensagens TC, que correspondem aos tipos 1 e 2, respectivamente.

O campo *Vtime* representa o tempo de validade dos dados contidos na mensagem, caso o nó não venha a receber nenhuma nova mensagem com atualizações. Os campos seguintes *Message Type*, *Originator Address* e *Time To Live* são, respectivamente, o tamanho da mensagem desde o campo tipo da mensagem até o fim do pacote ou o início da mensagem seguinte, o endereço IP de quem gerou a mensagem e o tempo de vida.

O campo *Hop Count* deve ser incrementado por todo nó antes de ser encaminhado e representa a distância em saltos percorrida pelo pacote. O campo de *Message Sequence Number* contém um número de sequência da mensagem tem por objetivo evitar que uma mensagem seja processada duas vezes pelo mesmo nó.

O campo *MESSAGE* contém as mensagens que são transmitidas pelo pacote do OLSR, onde cada pacote pode ter concatenado várias mensagens de maneira a diminuir a sobrecarga com pacotes de controle na rede.

A mensagem *HELLO* tem por objetivo tanto a descoberta de vizinhos e a sinalização da seleção de MPR, bem como a verificação da conectividade do enlace aos vizinhos. Essa mensagem não deve ser encaminhada pelos nós, por isso seu TTL (*Time to Live*) é colocado em 1. O formato da mensagem de *HELLO*, que corresponde ao campo *MESSAGE* do pacote na

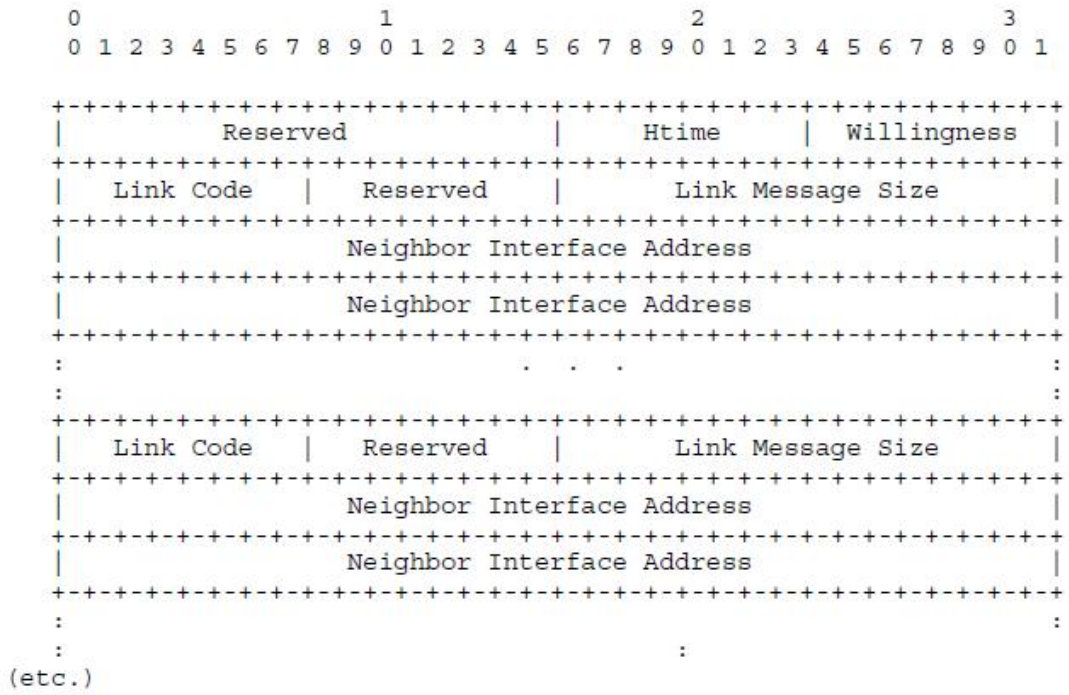

Figura 2.6: Mensagen *HELLO*

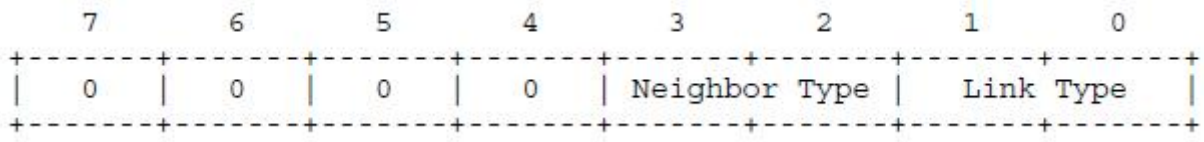

Figura 2.7: *Link Code*

Figura 2.5 é mostrado na Figura 2.6.

O campo *Htime* especifica o intervalo de emissão desse tipo de mensagem. O campo *Willigness* especifica se o nó pode (WILL_DEFAULT), não pode (WILL_NEVER) ou sempre (*WILL_ALWAYS*) será selecionado como MPR.

O campo *Link Code* é divido conforme mostrado na Figura 2.7, onde *Link Type* indica para cada enlace do emissor do *HELLO* com um vizinho:

- Symmetric: o enlace foi verificado como simétrico.
- Asymmetric: a comunicação só foi verificada em uma direção.
- Unspec: indica que nenhuma informação especifica sobre o enlace foi dada.
- Lost: Indica que o enlace foi perdido.

O campo *Neighbor Type* indica, para cada enlace com um vizinho do nó que enviou o *HELLO*, que pelo menos uma interface desse vizinho tem conexão simétrica com o emissor

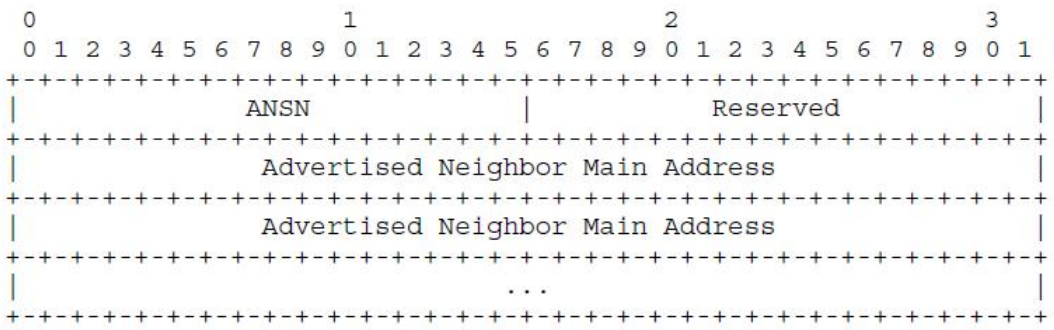

Figura 2.8: Mensagem TC

(*SYM_NEIGH*), ou ainda que o enlace é simétrico e selecionado como seu MPR (*MPR_NEIGH*), ou é assimétrico ou indisponível (*NOT_NEIGH*).

Ao receber uma mensagem *HELLO*, o nó deve atualizar sua tabela de enlaces, ou seja, uma tabela que contém os vizinhos distantes um salto, incluindo as informações sobre o tipo de enlace e validade daquela informação (campo *Vtime*).

Da mesma maneira, o nó também deve atualizar a tabela de vizinhos distantes um salto incluindo a informação relativa à sua voluntariedade (ser MPR ou não). Note que essa tabela é parecida com a anterior, já que ambas funcionam em par. Entretanto, a primeira guarda informações relativas ao estado do enlace, enquanto que a segunda sobre o vizinho. No caso desse vizinho ter selecionado este nó como MPR, o que é indicado pelo campo tipo de vizinho igual a *MPR_NEIGH*, ele deve ser adicionado à lista de selecionador MPR. Essa lista identifica que nós selecionamos este nó como MPR.

O nó deve processar também as informações relativas aos nós distantes dois saltos dele, ou seja, aqueles com os quais seus vizinhos possuem alcance direto, mas com os quais ele não possui. Assim, para cada endereço listado no campo endereço da interface do vizinho, o nó deve checar se este não é o próprio e adicioná-lo à tabela de vizinhos distantes de dois saltos, incluindo o endereço do vizinho emissor do pacote e o tempo de validade da informação.

Em todas as tabelas, de enlaces, de vizinhos distantes de um salto e de vizinhos distantes de dois saltos, a informação é excluída da tabela caso o seu tempo de validade expire ou o campo estado do enlace da mensagem *HELLO* indique que o enlace deixou de existir ou, ainda no último caso, que se tornou assimétrico. A construção da mensagem *HELLO* é feita de maneira análoga à recepção, onde cada nó deve enviar as informações referentes à sua tabela de enlaces, de vizinhos e de seus MPR.

Os mapas de topologia são construídos através da propagação das informações relativas aos vizinhos obtidas das mensagens *HELLO* e pelas mensagens de controle de topologia (*Topology Control Messages – TC messages*). Essas mensagens, cujo formato pode ser visualizado na Figura 2.8, são disseminadas pelos nós MPR, contendo informações suficientes para que cada nó construa suas tabelas de roteamento.

O campo ANSN (*Advertised Neighbor Sequence Number*) representa o número de sequência do anúncio. Toda vez que alguma mudança de topologia for detectada, esse número

Figura 2.9: Exemplo de entrada na tabela de rotas do OLSR

deve ser incrementado. Esse mecanismo permite que os demais nós saibam qual é a mensagem mais recente sobre a topologia da rede.

Cada campo *Advertised Neighbor Main Address* contém o endereço de um nó vizinho ao emissor do pacote. Esse nó deve ser distante um salto e devem ser enviados na mensagem pelo menos os endereços dos nós selecionados como MPR do nó originador do pacote.

A mensagem de controle de topologia deve ser enviada com TTL de valor máximo e deve ser propagada de maneira a atingir todos os nós da rede. Através da lista de vizinhos contidos no pacote, cada nó obtém todas as informações necessárias para construir sua tabela de roteamento. Mesmo quando a tabela de vizinhos estiver vazia, o nó deve ainda assim enviar a mensagem TC de forma a que os demais nós possam atualizar suas tabelas de roteamento.

Ao receber uma mensagem TC, caso o enlace não exista na tabela de topologia, o nó deve adicionar para cada vizinho anunciado uma entrada contendo o endereço do emissor do pacote, o número de sequência (ANSN) e o tempo de validade da informação. A mensagem TC pode conter como redundância outros nós que não os MPR, o que ajudaria a criar outras rotas para o tráfego de dados. Existem três parâmetros de configuração definido: o primeiro, no qual só os endereços dos MPR são incluídos nos vizinhos anunciados, o segundo, onde além dos MPR, os nós selecionados para serem MPR de outros nós são enviados e, finalmente, o terceiro modo de operação onde todos os vizinhos são incluídos.

2.1.3.3 Cálculo da Tabela de Rotas

A tabela de roteamento é construída com base nas informações contidas na tabela de enlaces, na tabela de topologia e nas tabelas de vizinhos distantes a um e dois saltos.

As tabelas de vizinhos distantes a um e dois salto são montadas com base nos dados presentes nas mensagens *HELLO*. As tabelas de topologia e de enlaces são construídas a partir das informações presentes nas mensagens TC, com base nas informações de distância, endereço do nó de origem da mensagem e do vizinho anunciado.

Cada tupla da tabela de roteamento, Figura 2.9, tem os seguintes campos:

- R_dest_addr: Endereço do destino da rota;
- R_next_addr: Endereço do próximo salto;
- R_dist: Distância em saltos até o destino;
- R_iface_addr: Interface de saída da rota.

Muitos nós podem ser utilizados como endereço do próximo salto (*R_next_addr*) para alcançar um destino (*R_dest_addr*), entretanto aqueles selecionados para serem MPR de outro nó são preferíveis como próximo salto. Pela forma como a sua rota é calculada, não é possível ter múltiplas rotas para um mesmo destino, uma vez que a cada vez que for calculada a tabela de roteamento, apenas uma rota para cada destino será obtida.

As entradas são gravadas na tabela de roteamento para cada destino na rede em que o caminho é conhecido. Todos os destinos, para os quais os caminhos estão interrompidos ou que seja conhecido "parcialmente", não são gravados na tabela de roteamento.

A tabela de roteamento é renovada quando ocorre uma mudança em:

- No conjunto de links,
- No conjunto de vizinhos,
- No conjunto de vizinhos a até dois saltos de distancia,
- Nas interfaces múltiplas associadas à base de informação.

A tabela de roteamento é recalculada no caso de um novo vizinho ou um vizinho perdido, quando um conjunto de nós a 2 saltos é criado ou removido, quando uma tupla de topologia é criada ou removida ou quando há mudança nas interfaces múltiplas associadas à base de informação. A renovação desta informação de roteamento não gera nenhuma mensagem a ser transmitida, nem na rede, nem na vizinhança a 1 salto de distância.

Muitos nós podem ser utilizados como endereço do próximo salto (*R_next_addr*) para alcançar um destino (*R_dest_addr*), entretanto aqueles selecionados para serem MPR de outro nó são preferíveis como próximo salto. Pela forma como a sua rota é calculada, não é possível ter múltiplas rotas para um mesmo destino, uma vez que a cada vez que for calculada a tabela de roteamento, apenas uma rota para cada destino será obtida.

O algoritmo de cálculo de rotas do OLSR segue os seguintes passos:

1. Todas as entradas da tabela de rotas são removidas;
2. As novas entradas na tabela de roteamento são registradas iniciando com vizinhos simétricos a 1 salto como nós destinos;
   (a) Para cada registro na tabela de vizinhos a 1 salto, cujo enlace seja simétrico, uma nova entrada de rota é registrada na tabela de roteamento. O destino (R_dest_addr) e o endereço do próximo salto (R_next_addr) são iniciados com o endereço do vizinho analisado, e a distância (R_dist) é iniciada com 1 salto.
3. Para cada nó N a 2 saltos, tal que exista pelo menos um vizinho V a 1 salto, onde V também é vizinho de N, é registrado na tabela de rotas o destino como sendo o nó N, a dois saltos, o próximo salto será o vizinho V, a um salto e a distância será iniciada com 2 saltos.

4. Para cada nó N distante a mais de 2 saltos, é procurado um destino D cuja a rota para ele já se encontre na tabela de rotas. Caso se encontre, será criada uma entrada para o destino N na tabela de rotas onde o próximo salto será o mesmo para se chegar a D e a distância será o número de saltos para se alcançar D + 1. Caso se encontre mais de uma rota para o destino, é verificado qual tem o menor número de saltos e a qual é melhor avaliada segundo alguma métrica de roteamento, a melhor das duas será a rota escolhida.

Como se observa no algoritmo anterior, por fazer busca em largura, o OLSR calcula rotas primeiro para destinos a 1 salto de distância, depois a 2 saltos e então os destinos a mais de 2 saltos, até que seja construída a rota para o último destino mais distante da rede.

2.1.3.4 Métrica de Roteamento ETX

A métrica ETX (*Expected Transmission Count*) visa minimizar o número esperado de tentativas de transmissão para uma transmissão com sucesso, incorporando os efeitos de taxa de perda, assimetria nas taxas de perda nas duas direções de um enlace e interferência ao longo dos sucessivos enlaces de um caminho. Nos protocolos de roteamento para MANET, mencionados anteriormente, o melhor caminho é aquele que minimiza a métrica número de saltos, a despeito da vazão em todos os caminhos possíveis.

Utilizando simplesmente a métrica número de saltos, os protocolos assumem que cada enlace ou está funcionando ou está inoperante e neste caso simplesmente não há nenhuma comunicação. Embora essa suposição seja muito próxima da realidade para redes cabeadas, não é o caso das redes sem fio. Enlaces com uma elevada taxa de perda de pacotes dados, entregando menos de 50% dos pacotes, mas ainda sim transportando mensagens de controle, seriam considerados da mesma maneira que enlaces com boa vazão.

A proposta da métrica ETX é encontrar caminhos que produzam o menor número necessário de retransmissões. Para isso a métrica prevê o número de retransmissões esperadas fazendo medições da taxa de perda de pacotes para cada enlace da rede. O objetivo é, assim, encontrar rotas com a melhor vazão.

Para tanto, se o ETX de um enlace é o número de transmissões necessárias para enviar um pacote sobre ele, o ETX de uma rota será a soma dos ETX de cada enlace. Em um teste descrito em (BOUKERCHE, 2004), usando ETX com os protocolos DSR e *Destination-Sequenced Distance Vector* (DSDV), foram obtidas vazões melhores por uma razão de dois com o uso do ETX. Os resultados apresentados são melhores à medida que o número de saltos aumenta, mostrando a eficácia da métrica conforme a rede cresce.

No caso do protocolo OLSR a solução adotada foi diferente da vista em (ALEXANDER, 2004), uma vez que para medir a taxa de erro foi usado a própria mensagem *HELLO*, que já é transmitida periodicamente na rede, ao invés de se criar outro tipo de mensagem, evitando assim o aumento da sobrecarga de controle da rede. Uma desvantagem de seu emprego, é que sempre será enviado um pacote de tamanho pequeno, uma vez que a mensagem *HELLO* possui essa característica.

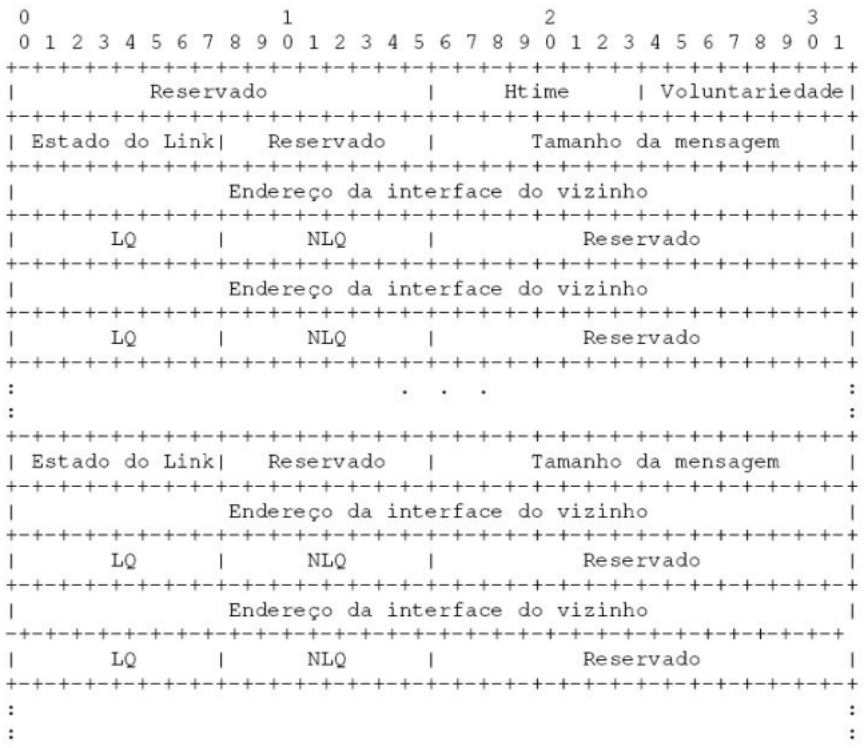

Figura 2.10: Mensagem LQ HELLO

Como cada nó sabe que as mensagens *HELLO* são transmitidas periodicamente, ele pode medir a taxa de erro relacionando a quantidade de pacotes recebidos com a quantidade de pacotes esperados em uma determinada janela de tempo.

Por exemplo, se 3 em cada 10 mensagens *HELLO* não forem recebidas pelo vizinho, tem-se uma taxa perda de 3/10 = 30%. Assim, como 7 mensagens chegaram tem-se uma qualidade de enlace (*Link Quality* – LQ) de 70%.

Entretanto, como os enlaces são bidirecionais, cada nó também obtém informações sobre a qualidade do enlace relativa aos pacotes que ele está enviando, ou seja, a visão que os vizinhos têm de seu enlace (*Neighbor Link Quality* – NLQ). Como ambos esses valores são enviados em percentagem, eles representam a probabilidade de um pacote atravessar um enlace com sucesso em cada direção. Assim, a probabilidade de sucesso de uma transmissão será o produto dessas: LQ x NLQ.

Pode-se concluir então que o número de tentativas esperado para que um pacote possa ser transmitido com sucesso pode ser visualizada na equação 2.1.

$$ETX = \frac{1}{LQL * NLQ} \tag{2.1}$$

É importante notar que esse valor é válido nos dois sentidos do trajeto, uma vez que haverá retransmissão tanto se o pacote de dados quanto se o respectivo ACK forem perdidos.

Finalmente, conforme mencionado anteriormente, o ETX de uma rota será a soma

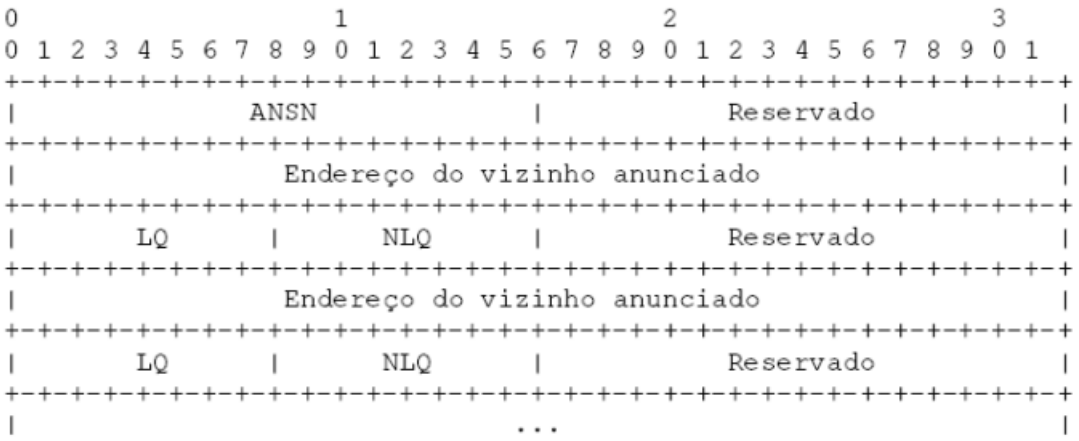

Figura 2.11: Mensagem LQ TC

da métrica de cada enlace dos nós intermediários. Para incluir a informação de qualidade do enlace nas mensagens *HELLO* do OLSR, elas foram modificadas como proposto na RFC 3626 e mostrado anteriormente. Conforme a Figura 2.10, a nova mensagem chamada de LQ *HELLO* contém agora dados sobre a qualidade do enlace.

Após o endereço do vizinho foi incluída a informação relativa à qualidade do enlace nos dois sentidos.

Para transmitir essa informação para todos os vizinhos para que eles a considerem no cálculo da rota, também foi necessário alterar a mensagem de controle de topologia, que foi denominada LQ TC, e cujo formato é apresentado na Figura 2.11.

Da mesma maneira que na LQ *HELLO*, informações relativas à qualidade do enlace do nó que está enviando a mensagem com o respectivo vizinho são incluídas logo após seu endereço.

Para a construção da tabela de roteamento utilizando a qualidade da métrica é empregado o algoritmo de *Dijkstra*, onde é procurado em toda a topologia o caminho de menor custo entre dois nós.

2.1.4 AODV

O AODV (PERKINS; BELDING-ROYER; DAS, 2003) é um protocolo de roteamento reativo baseado no algoritmo de vetor de distância (KUROSE; ROSS, 2009) para obter um melhor desempenho em rede sem fio.

O AODV tenta eliminar a necessidade de um broadcast global das informações de roteamento, evitando o alto consumo de banda por pacotes de controle. O AODV possui algumas características que possibilitam uma rápida adaptação a modificações de topologia.

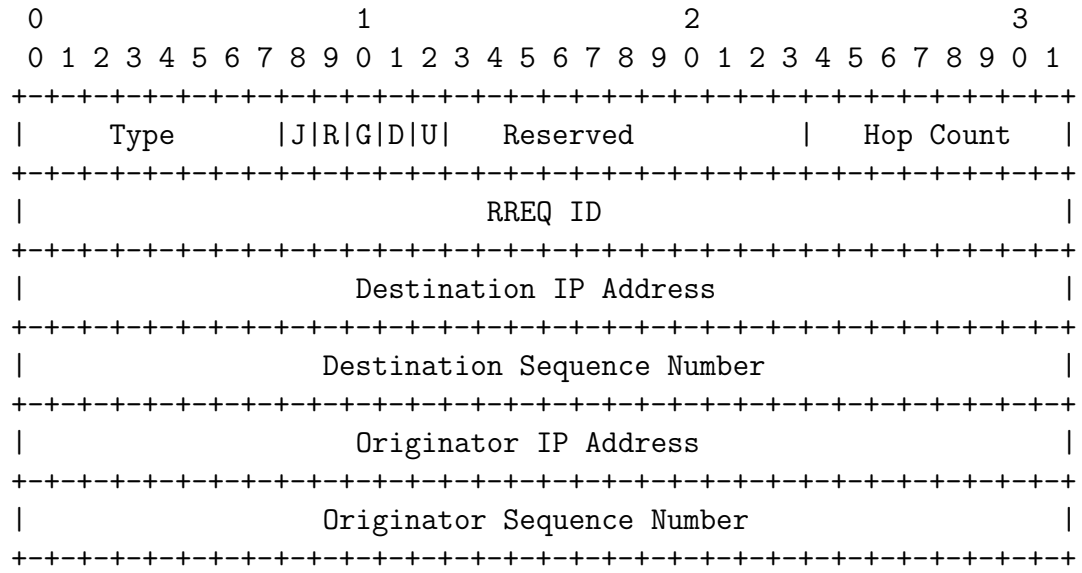

Figura 2.12: Formato do pacote de RREQ do protocolo AODV

2.1.4.1 Descoberta de Rotas

Quando um nó possui a necessidade de enviar informações para outro nó da rede e não identifica uma rota válida em sua tabela de roteamento para alcançar o nó destino, então ele envia um pacote de requisição de rotas chamado de *Route REQuest* (RREQ), Figura 2.12, em *broadcast*.

Cada RREQ possui um identificador (RREQ ID), que junto com o endereço IP de origem, serve para diferenciar uma mensagem RREQ específica de outras mensagens RREQs. O RREQ ID é incrementado pelo nó origem para cada RREQ, o que torna o identificador único.

Quando um nó intermediário recebe uma mensagem RREQ, eles verificam se são o destino ou se conhecem uma rota para o destino. No caso do nó ser o destino, ou de possuir em sua tabela de roteamento o caminho para o nó destino, o mesmo envia uma mensagem de resposta de rota *Route REPly* (RREP), Figura 2.13 para o nó de origem da requisição, pelo mesmo caminho no sentido oposto, mantendo o número de sequencia do pacote RREQ. Caso não seja o destino e não conheça o caminho para o destino, o nó intermediário encaminha a solicitação adiante aos seus nós vizinhos.

A tabela de roteamento do AODV é formada pelos seguintes campos:

1. endereço de destino;
2. número de sequência do destino;
3. interface de rede;
4. número de saltos (quantidade de saltos para chegar ao nó destino);

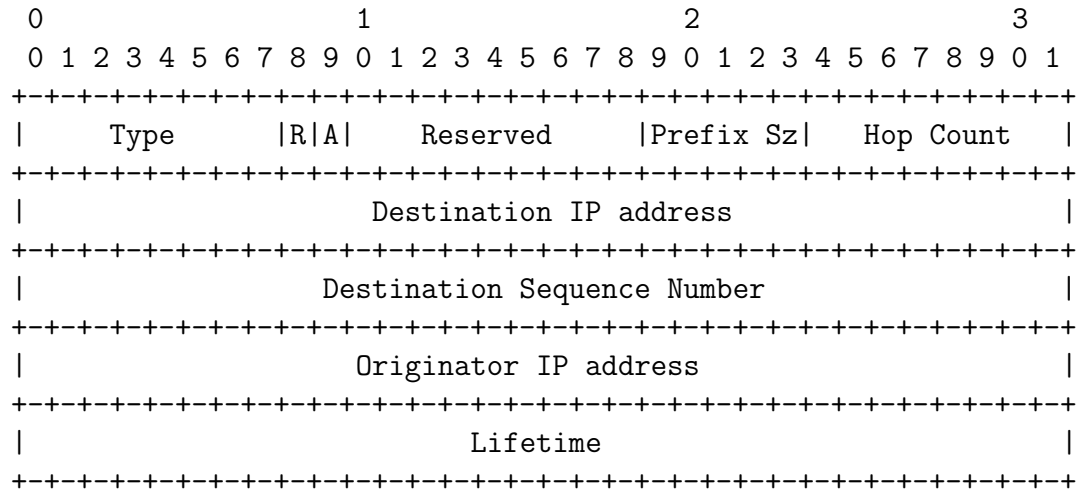

Figura 2.13: Formato do pacote de RREP do protocolo AODV

5. próximo salto;
6. lista de precursores (nós que utilizam a rota);
7. tempo de vida da rota.

Cada nó conhece apenas o próximo salto e a quantidade de saltos para se alcançar o destino. Cada nó possui seu número de sequência, o qual é incrementado a cada mensagem RREQ, RREP ou RERR, enviada quando ocorre algum erro em uma rota. Este número de sequência serve para identificar a rota mais recente. Um nó sabe que o número maior é o mais recente, portanto, não propaga uma rota com um número menor do que uma rota que este já tenha propagado. Além disso, permite identificar se uma requisição de rota chegou duplicada. Em caso afirmativo, o nó pode desconsiderar o último pacote de broadcast e evitar processamento e encaminhamento desnecessário do pacote em questão.

2.1.4.2 Manutenção de Rotas

O mecanismo de manutenção de rotas do AODV possui dois mecanismos básicos:

1. descobertas de rotas com erro;
2. descarte de rota com tempo de vida esgotada.

Para a identificação rápida de quebra de enlace, mensagens *HELLO* são enviadas periodicamente de cada nó para seus vizinhos. Se ocorrer de um nó não receber mais mensagens *HELLO* de um vizinho, o mesmo assume que aquele enlace não está mais ativo e dissemina essa informação para os nós que dependiam dele para estabelecer determinadas rotas. Esses nós se encontram listados na lista de precursores, campo associado a cada rota na tabela de

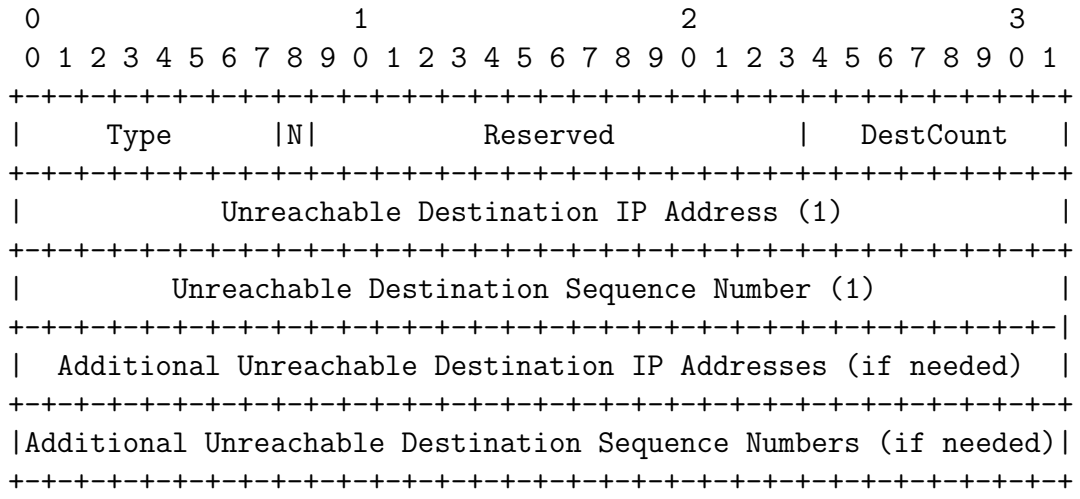

Figura 2.14: Formato do pacote de RRER do protocolo AODV

roteamento. Este aviso é realizado pela mensagem de erro de rota *Route Error* (RERR), Figura 2.14. Ela possui um campo que representa o número de saltos, esse campo recebe um valor extremamente alto, simbolizando que o enlace se encontra foca de alcance. Um nó ao receber uma mensagem RERR pode utilizar o mecanismo de requisição de rotas para buscar outro caminho, ou simplesmente interromper a comunicação com o nó destino.

O segundo mecanismo de manutenção de rotas verifica se o tempo de vida da rota se encontra esgotado, caso sim a rota é descartada.

2.1.5 AOMDV

O AOMDV (Ad hoc On Demand Multipath Distance Vector) (NGUYEN et al., 2004) é uma extensão do AODV cuja a principal característica é a capacidade de computar múltiplas rotas durante a fase de descoberta de rotas. Ele foi desenvolvido para ser utilizado em redes Ad Hoc com alto dinamismo onde ocorrem frequentemente falhas de enlaces e quebras de rotas.

No AODV quando ocorre a quebra de uma rota no meio de uma transmissão de dados é necessário executar uma nova chamada para o mecanismo de descoberta de rotas. Esse procedimento é frequentemente associado com uma alta latência e uma alta sobrecarga da rede por mensagens de controle. Utilizando múltiplos caminhos, caso alguma rota caia, basta concentrar a transmissão de dados em uma outra rota disponível, o mecanismo de descoberta de rotas só será chamado em caso de todas as rotas disponíveis se tornarem invalidas, (NGUYEN et al., 2004).

2.1.5.1 Descoberta e Manutenção de Rotas

O mecanismo de descoberta de rotas do AODV foi alterado para permitir o processamento de múltiplas rotas disjuntas por enlace. No AOMDV, tal qual o AODV, quando um nó

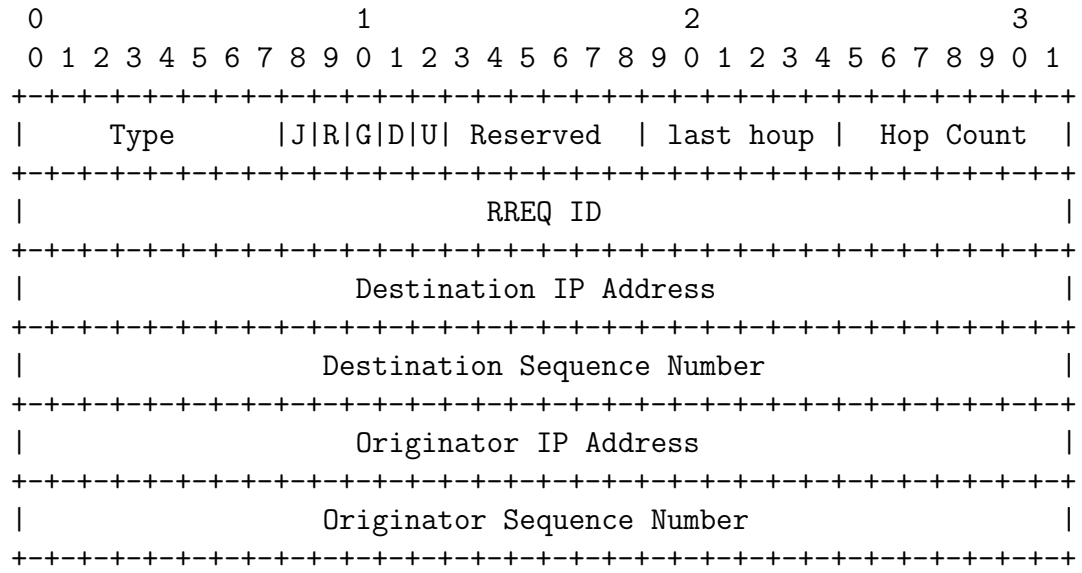

Figura 2.15: Formato do pacote de RREQ do protocolo AOMDV

necessita de uma rota para um destino e não encontra nenhuma referência válida para a mesma na sua tabela de rotas, tem início o mecanismo de descoberta de rotas.

O processo de descoberta de rotas consiste em transmitir um pacote de requisição de rotas RREQ pela rede até atingir o nó destino ou ser respondido por um nó intermediário que contenha uma rota para o destino em sua tabela de rotas. Em qualquer um dos casos é gerada uma mensagem RREP como resposta para a mensagem RREQ. Essa mensagem fará o caminho inverso ao percorrido pela mensagem RREQ até alcançar o nó origem.

Cada mensagem RREQ do AOMDV possui um campo a mais chamado de *firsthop* definido como o primeiro salto do nó, ou seja, um nó vizinho do nó origem por onde a mensagem passou inicialmente. Essa informação será armazenada em uma lista presente em cada nó chamada de *firsthop_list*. Quando um nó recebe um RREQ duplicado ele não o descartará imediatamente como o AODV, primeiro ele verificará se o campo *firsthop* do nó já se encontra na sua *firsthop_list*. Caso o campo *firsthop* não se encontre na *firsthop_list*, a mensagem RREQ poderá ser ou respondida, caso o nó possua algum caminho para o nó destino em sua tabela de rotas ou seja o nó destino, ou encaminhada para o seus vizinho. Caso o campo *firsthop* se encontre na *firsthop_list* essa mensagem será descartada. Dessa forma pelo menos um nó no caminho será diferente, garantindo assim a disjunção por enlace.

A tabela de roteamento do AOMDV é formada pelos seguintes campos:

1. endereço de destino;
2. número de sequências do destino;
3. número de saltos anunciados;
4. lista de precursores (nós que utilizam a rota);

5. lista de rotas. (lista com as rotas descobertas para o destino)

   (a) número de saltos (quantidade de saltos para chegar ao nó destino);

   (b) próximo salto;

   (c) interface de rede;

   (d) tempo de vida da rota.

Dentre os campos apresentados na tabela de rotas se destacam o ”número de sequencia do destino”, e o ”número de saltos anunciados”. Da mesma forma do AODV, no AOMDV todos os nós possuem um número de sequência. Esse valor impede a retransmissão de mensagens repetidas e ajuda aos protocolos a manterem suas tabelas de roteamento atualizadas.

O campo ”número de saltos anunciados”representa o maior número de saltos pertencente a um caminho presentes na ”lista de rotas”para um dado destino. Após definido esse valor só serão aceitas rotas com o mesmo número de frequência para o destino caso o número de saltos seja menor que o valor do campo ”número de saltos anunciados”. Esse mecanismo previne a formação de rotas circulares. Ele é anunciado para todos os nós presentes na lista de ”lista de precursores”. O valor desse campo só será alterado em caso de mudança do número de sequencia do nó destino.

2.1.5.2 Manutenção de Rotas

A manutenção de rotas no AOMDV é uma extensão da manutenção de rotas do AODV. Como no AODV, o AOMDV também usa pacotes RERR para indicar enlaces comprometidos. Caso essa verificação ocorra no meio de uma transmissão, automaticamente o nó intermediário encaminha o fluxo de dados para uma rota alternativa.

O mecanismo de *timeout* das entradas na tabela de rotas do AODV também foi alterado. Ao invés de um temporizador por entrada na tabela de rotas, passou a ser usado um para cada rota. A determinação de cada valor é mais difícil no AOMDV em comparação ao AODV. Como existem vários caminhos é mais provável um deles se tornar obsoleto. Porém não é aconselhado fazer uso de valores pequenos para o *timeout* pois pode ocorrer de caminhos viáveis serem descartados ocasionando a perda de possíveis caminhos.

2.2 RTP/RTCP

A RFC 3550 (SCHULZRINNE et al., 2003) define dois protocolos o *Real-Time Transport Protocol* (RTP) e o *Real-Time Transport Control Protocol*(RTCP). O primeiro é utilizado para transmissão de dados em tempo real e o segundo é responsável por gerar as informações de controle para a sessão do primeiro.

O RTP possui características tanto de um protocolo de camada de sessão, como de um protocolo entre as camadas de sessão e de transporte (KUROSE; ROSS, 2009). Seu objetivo é

prover serviços de transporte de dados fim-a-fim em tempo real, como áudio e vídeo, em redes IP unicast e multicast.

Em uma comunicação IP, cada participante da sessão envia continuamente um fluxo de dados em trechos de aproximadamente 20ms encapsulado num cabeçalho RTP. Esses cabeçalhos são encapsulados por um protocolo da camada de transporte, normalmente é utilizado o protocolo UDP, pois este gera menos atrasos devido a não executar procedimentos de garantia de entrega como o TCP, entretanto nada impede o uso do RTP com o TCP.

Ao chegarem ao destino, os pacotes são armazenados e organizados no chamado *buffer* de *playout* (ou *jitter buffer*). O *buffer* de *playout* tem o objetivo de garantir a reprodução síncrona e contínua da conversa entre os participantes de uma sessão RTP. Assim ele serve para absorver as variações do atraso dos pacotes e descartar aqueles fora do seu intervalo de tempo de reprodução, ou seja, fora do seu *playout*. Assim percebe-se uma relação direta entre o atraso e a perda de pacotes RTP (MARCONDES et al., 2003).

O RTP não possui nenhum mecanismo para averiguar se a transmissão esta ocorrendo da forma desejada. Para esse fim foi criado o *Real-Time Transport Control Protocol*(RTCP). O RTCP se baseia em transmissões periódicas de pacotes de controle, chamados de relatórios RTCP, entre os participantes de uma sessão RTP. O RTCP executa quatro funções:

1. a função básica é fornecer informações sobre a qualidade da distribuição dos dados. A emissão de relatórios a todos os participantes permite o diagnóstico de problemas tanto locais como globais. Esta função é executada pelos relatórios do remetente (SR) e do receptor (RR) de RTCP, esse pacotes serão descritos adiante;

2. o RTCP carrega um identificador, persistente ao nível de transporte, para uma fonte de RTP chamado Nome Canônico (CNAME). Visto que o identificador SSRC pode mudar em caso de conflito ou se um programa for reiniciado, os receptores precisam do CNAME para cada um manter informações de cada participante e para poder associar a estes diversos fluxos, em várias sessões RTP relacionadas;

3. as primeiras duas funções requerem que todos os participantes enviem pacotes RTCP, portanto a frequência de envio destes deve ser controlada para que o RTP seja escalável para um grande número de participantes. Como cada participante envia seus pacotes de controle a todos os outros, cada um pode independentemente observar o número total dos participantes. Este número é usado calcular a taxa em que os pacotes são enviados;

4. uma quarta função, que é opcional, é a de transmitir informações mínimas de controle da sessão, como por exemplo, a identificação dos participantes seja mostrada na interface do usuário.

A taxa de transmissão de dados do RTCP é controlada para não comprometer a banda de dados com mensagens de controle. O intervalo de envio de mensagens para os emissores segue a fórmula 2.2 e o intervalo de envio para os receptores segue a fórmula 2.3.

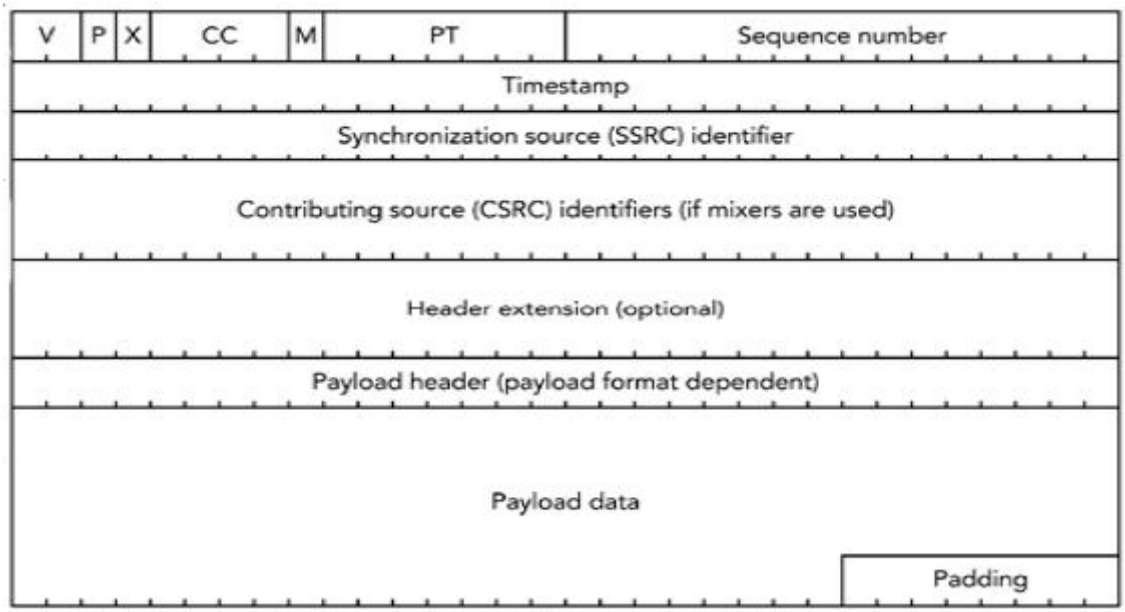

Figura 2.16: Pacote RTP

$$T\_emi = \frac{numero\_emissores}{0,25 * 0,05 * banda\_sessao}(dim\_media\_pac\_rtcp) \tag{2.2}$$

A RFC 3550 recomenda o uso de no máximo 5% da banda disponível para a transmissão de relatórios RTCP de emissores e que dessa banda até 25% seja dedicada para a transmissão dos outros participantes da rede. Para os pacotes RTCP enviados pelos receptores o máximo da banda utilizada também deve ser de 5% e a fração dessa banda reservada para a emissão dos outros participantes deve ser de até 75%.

$$T\_recep = \frac{numero\_emissores}{0,75 * 0,05 * banda\_sessao}(dim\_media\_pac\_rtcp) \tag{2.3}$$

As variáveis "numero_emissores", "dim_media_ pac_rtcp"e "banda_sessao"representam respectivamente o total de emissores participantes da sessão RTP, uma estimativa da média de tamanho dos pacotes agregados para adaptar a taxa de transmissão a quantidade de informações a serem enviadas e o tamanho da banda disponível para a sessão.

2.2.1 Pacote RTP

- V - Versão: Campo utilizado para identificar a versão do protocolo;
- P: Utilizado para enchimento (*padding*). Se o bit esta setado, o pacote contém alguns octetos de enchimento no final do pacote. Esse espaço pose ser utilizado por algum algoritmo de criptografia;
- X: Utilizado para extensão. Se o bit está ligado, o cabeçalho fixo e seguido por uma extensão;

- CC - Contador CSRC: Contém o número de identificadores CSRC que seguem o cabeçalho fixo;
- M - *Marker*: Utilizado para marcação. A interpretação do marcador é definida por um perfil. Um perfil pode definir bits adicionais de marcação ou até mesmo excluir esses bits existentes no cabeçalho padrão. Geralmente esse bit e usado para sinalizar algum evento importante como, por exemplo, o término de um frame;
- PT - Tipo de *payload*: Esse campo identifica o formato da área de dados do RTP e sua interpretação pela aplicação. Alguns *payload*s são definidos na RFC (GROUP et al., 1996) e na RFC *Assigned Numbers*, como por exemplo, para o H.323 a referência é o H.225;
- Número de Sequencia: O número de sequência é incrementado em uma unidade quando um pacote de dados RTP é enviado. Pode vir a ser usado pelo nó destino para detectar perdas de pacotes e para reordenar os pacotes recebidos. O valor inicial do número de sequência é atribuído aleatoriamente;
- *Timestamp*: O *timestamp* reflete a amostra do primeiro octeto de dados RTP. Essa amostra é derivada de um relógio que é incrementado linearmente com o intuito de prover sincronização e cálculo da variação do atraso (*jitter*) ;
- Identificador de fonte de sincronização SSRC *(Synchronization Source Identifier)*: Esse campo identifica a origem da sincronização. Seu valor é escolhido aleatoriamente para não ocorrer códigos duplicados em uma mesma sessão RTP;
- Identificadores de fonte contribuinte CSRC *(Contributing Source Identifiers)*: Normalmente os dados RTP são gerados por apenas uma fonte, porém quando existe mais de uma, os dados passam por uma entidade chamada *mixer*, responsável por juntar os fluxos em um só. Nesse campo se encontrará o seu identificador. Cada identificador é composto por 32 bits, correspondente ao SSRC do transmissor. O tamanho do CSRC e definido pelo campo CC.

2.2.2 Pacote RTCP

Os pacotes especificados pelo RTCP são:

- receiver reports (RR)
- sender reports (SR);

O primeiro contém informações a respeito das métricas de qualidade de transmissão observadas pelo o nó destino. O segundo contém informações a respeito da transmissão em execução pelo nó origem.

A Figura 2.17 representa o cabeçalho base de todo pacote RTCP. Ele é formado pelos seguintes campos:

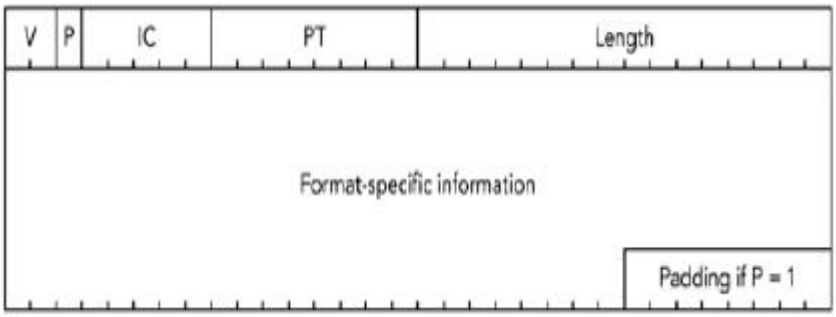

Figura 2.17: Cabeçalho base do RTCP

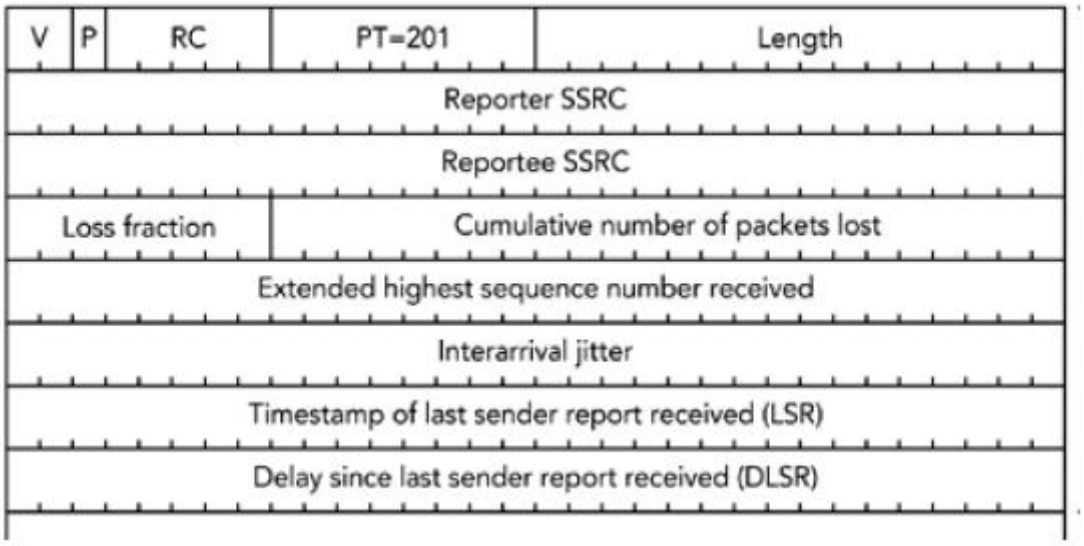

Figura 2.18: Pacote RTCP *Receiver Report*

- V: Versão do RTP;
- P: Utilizado para enchimento (*padding*). Se o bit esta setado, o pacote contém alguns octetos de enchimento no final do pacote. Esse espaço pose ser utilizado por alguns algoritmos de criptografia;
- IC: Contador de itens (*Item Count*) Indica quantos itens existem nos relatórios;
- PT: Informa o tipo do pacote (*Packet Type*). Caso o pacote seja um *Sender Report* o valor do PT será 200, caso seja um *Receiver Report* o valor será 201.
- Tamanho: Tamanho total do pacote.

2.2.2.1 RTCP RR - *Receiver Report*

A Figura 2.18 representa o pacote RTCP *Receiver Report*. Ele é formado pelos seguintes campos:

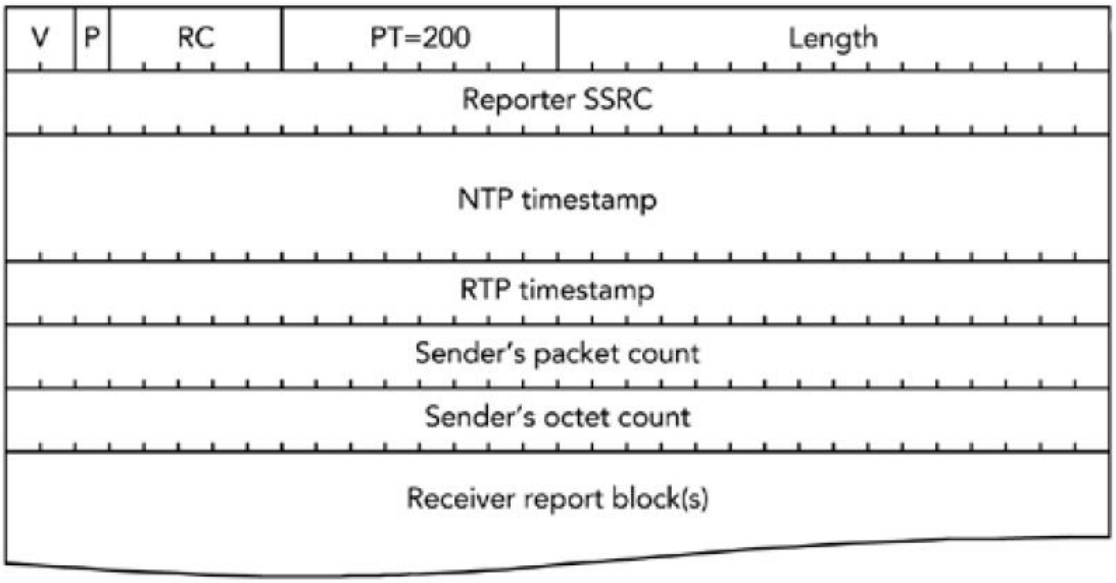

Figura 2.19: Pacote RTCP *Sender Report*

- *Reporter SSRC*: SSRC do participante que esta enviando o pacote;
- *Reportee SSRC*: Identifica de qual participante as estatísticas de qualidade do relatório estão se referindo;
- Fração de Perda: Valor do resultado dos (pacotes perdidos)/(pacotes enviados) multiplicado por 256;
- Pacotes perdidos acumulados: Número de pacotes perdidos desde o início da sessão;
- Valor estendido do mais alto número de sequência recebido: Os 16 bits mais significativos contêm o número de ciclos de sequencia (isto é, o número de vezes que a contagem atingiu o valor máximo e retornou a zero) e os últimos 16 bits contêm o mais alto número de sequência recebido em um pacote de dados RTP proveniente dessa fonte (mesmo SSRC);
- *Jitter* entre chegadas: É a diferença entre os tempos de chegada dos últimos pacotes;
- Último *timestamp* SR (LSR): Os 32 bits do meio do timestamp do ultimo Sender Report, representa o tempo de chegado do último relatório SR;
- Atraso desde o último SR (DLSR): Tempo desde que o último pacote SR chegou.

2.2.2.2 RTCP SR - *Sender Report*

A Figura 2.18 representa o pacote RTCP *Sender Report*. Ele é formado pelos seguintes campos:

- NTP *timestamp*: Valor no formato NTP representando o instante do envio do pacotes;

- RTP *timestamp*: Valor no formato NTP medido a partir da taxa de amostragem da mídia;
- Contagem de pacotes enviados: Quantidade de pacotes enviados desde o inicio da sessão;
- Contagem de octetos enviados: Quantidade de octetos enviados.

3 OLSR MCTR

Neste capítulo será apresentado o funcionamento do OLSR *Multipath for Real Time* (OLSR MCTR). A seção 3.1 é responsável por detalhar o funcionamento do OLSR MCTR. A seção 3.2 apresenta o algoritmo de cálculo de rotas o OLSR MCTR. A seção 3.3.1 descreve o funcionamento do procedimento responsável por calcular o número ideal de rotas. A seção 3.3 descreve o comportamento do OLSR MCTR quando ele transmite pacotes RTP.

3.1 Introdução

O OLSR MCTR não é um novo protocolo de roteamento, mas uma extensão do OLSR. Ele segue os princípios básicos de funcionamento desse protocolo, utilizando as mesmas mensagens, estruturas de dados nele implementados. Por herdar tais características do OLSR, o OLSR MCTR é, também, um protocolo de roteamento proativo baseado em estado de enlace.

Tanto no OLSR como no OLSR MCTR são mantidas entradas na tabela de roteamento para cada destino da rede. Por fazerem roteamento salto-a-salto, uma entrada na tabela de roteamento não registra toda rota até ao destino, mas apenas o endereço do próximo salto para alcançá-lo. Em redes MANETs, em situações de alta densidade de nós na rede, é possível fazer uso de vários nós como próximo salto na rota, pois fazem parte de outras possíveis rotas que levam para o mesmo destino.

Como é possível ver no trabalho de MAO et al. (2006), o uso de múltiplos caminhos melhora o desempenho de transmissões de dados em tempo real em redes Ad Hoc utilizando o protocolo RTP. Porém como Lu et al. (2008) comenta, o uso de vários caminhos nem sempre acarreta em benefícios para aqueles que os utilizam.

Esse trabalho propõe um novo mecanismo de cálculo de rotas para o OLSR e um novo método para encaminhar pacotes RTP. O OLSR MCTR é capaz de calcular múltiplas rotas para os destinos e de fazer uso das informações presentes nos relatórios RTCP RR (SCHULZRINNE et al., 2003) para definir quantos caminhos devem ser utilizados e quando é mais apropriado realizar a atualização da tabela de rotas durante uma transmissão.

O principal motivo para se utilizar essa estratégia é permitir ao protocolo de roteamento verificar se a transmissão esta ocorrendo dentro de intervalos de atraso fim-a-fim e de variação de atraso exigidos para a transmissão de dados em tempo real. Caso não esteja, é possível que o caminho utilizado se encontre congestionado ou já não seja mais viável, então o protocolo

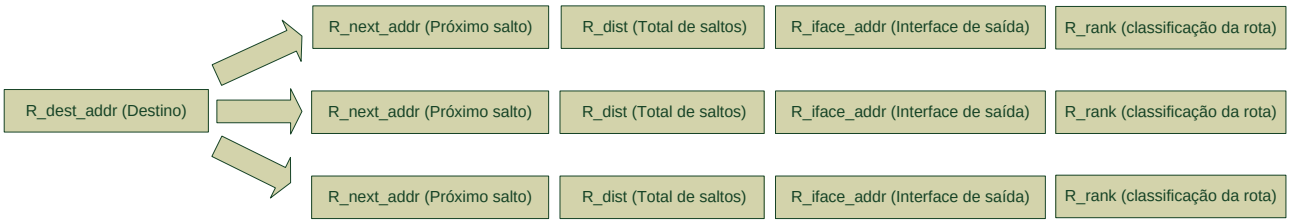

Figura 3.1: Exemplo de entrada na tabela de rotas do OLSR MCTR

utilizará outros caminhos com o intuito de realizar a transmissão de forma adequada. Caso o protocolo esteja transmitindo os dados dentro de intervalos apropriados, o novo mecanismo garante que a transmissão não seja interrompida para se iniciar a atualização da tabela de rotas.

A implementação do OLSR MCTR envolve a alteração de algumas estruturas de dados e procedimentos presentes no OLSR. Essas alterações serão melhor descritas nas próximas seções.

3.2 Cálculo de Rotas

No OLSR a tabela de roteamento é calculada toda vez que a rede é inicializada ou recalculada toda vez que ocorrem alterações na topologia da rede. A sua tabela de roteamento é capaz de armazenar apenas um caminho para cada destino.

O OLSR MCTR implementa uma tabela de rotas capaz de armazenar mais de uma rota para cada destino, conforme podemos observar na Figura 3.1. Cada entrada na tabela aponta para uma lista de caminhos possíveis. Um novo campo foi adicionado na tabela chamado de R_rank, ele é utilizado para inferir a prioridade de cada rota. O valor de prioridade é calculado em função da métrica de roteamento ETX, definida no Capítulo 2, dessa forma é possível ordenar os rotas em função da qualidade de seus enlaces. Os campos da tabela de roteamento do OLSR MCTR são descritos a seguir.

1. R_next_addr - indica o próximo salto;
2. R_dist - distância em saltos até o destino;
3. R_iface_addr - endereço da interface de saída;
4. R_rank - classificação da rota.

Conforme proposto na RFC 3626 (CLAUSEN; JACQUET, 2003) o cálculo da tabela de roteamento pelo OLSR é feito por meio da execução de um algoritmo de “caminho mais curto”. Após encontrar o primeiro “caminho mais curto” o vizinho simétrico do nó local, participante dessa rota, é eleito como o “próximo salto” (R_next_addr) em direção ao destino. No

OLSR MCTR, também é utilizado o algoritmo do “caminho mais curto”, porém, diferentemente do OLSR, o OLSR MCTR armazena todos os caminhos disponíveis e não somente um.

Todo o procedimento para calcular ou recalcular a tabela de roteamento de um nó executado pelo OLSR MCTR segue basicamente os mesmos princípios de execução do OLSR. A seguir é apresentado o procedimento utilizado para o cálculo da tabela de roteamento do OLSR MCTR.

1. Todas as entradas da tabela de rotas são removidas;
2. As novas entradas na tabela de roteamento são registradas iniciando com vizinhos simétricos a 1 salto como nós destinos;
   (a) Para cada registro na tabela de vizinhos a 1 salto, cujo enlace seja simétrico, uma nova entrada de rota é registrada na tabela de roteamento. O destino (R_dest_addr) e o endereço do próximo salto (R_next_addr) são iniciados com o endereço do vizinho analisado, a distância (R_dist) é iniciada com 1 salto, o valor de prioridade da rota (R_rank) será atualizada a partir da análise da métrica ETX.
3. Para cada nó N a 2 saltos, tal que exista pelo menos um vizinho V a 1 salto, onde V também é vizinho de N, é registrado na tabela de rotas o destino como sendo o nó N, a dois saltos, o próximo salto será o vizinho V, a um salto, a distância será iniciada com 2 saltos e o valor de prioridade da rota será atualizada a partir da análise da métrica ETX.
   (a) Caso seja encontrado mais de um caminho em direção ao mesmo destino, é criado mais uma entrada na lista de caminhos para o destino, onde um novo vizinho será tratado como próximo salto e a distância será iniciada com 2 saltos.
4. Para cada nó X na tabela de topologia, que não esta nem na tabela de vizinhos a 1 salto e nem na tabela de vizinhos a 2 saltos, é procurado um nó destino D na tabela de rotas que seja seu vizinho. Caso D seja encontrado, será criado na tabela de roteamento um novo registro para um destino apontando para X, o próximo salto será o vizinho que da acesso a rota de maior priorida para D, a distância será a quantidade de saltos até D + 1 e o valor de prioridade da rota será atualizada a partir da análise da métrica ETX.
   (a) Caso seja encontrado um nó destino D2 que também seja vizinho de X, o mesmo procedimento será adotado.

3.3 Encaminhamento de Pacotes

No OLSR quando um nó deseja transmitir um fluxo de dados ele busca uma rota até o destino na tabela de rotas, em seguida verifica por qual vizinho o pacote deverá ser encaminhado e o envia. No OLSR MCTR esse mecanismo funciona de uma forma diferente. Primeiro o nó verifica de que tipo é o fluxo de dados, caso não seja RTP, dentre as possíveis rotas para se alcançar o destino, é escolhida aquela com maior prioridade e em seguida o fluxo segue

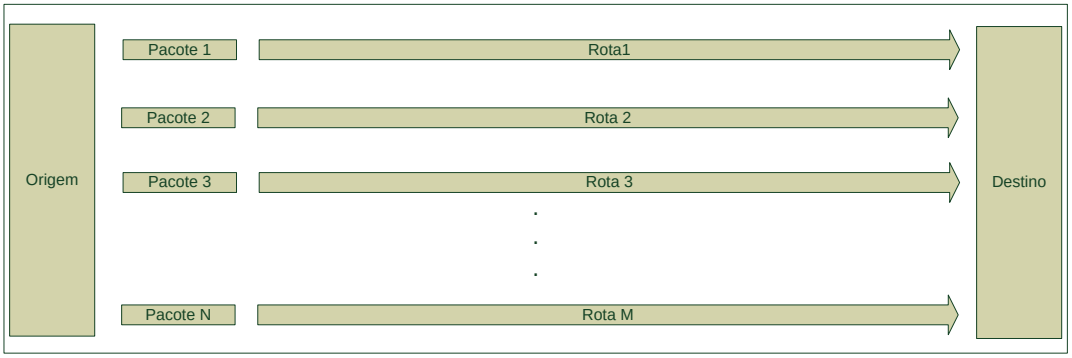

Figura 3.2: Distribuição alternada de N pacotes pelas M rotas

o mesmo procedimento executado pelo OLSR. Caso o fluxo seja RTP, o OLSR MCTR deve consultar as variáveis OLSR_MCTR_COUNT, RTABLE_SIZE e PATH_COUNT.

A variável OLSR_MCTR_COUNT define a quantidade máxima de caminhos que devem ser utilizados pelo nó para encaminhar o fluxo de dados. O seu valor é atualizada pelo algoritmo de cálculo do número ideal de rotas (NIR), esse algoritmo será descrito na próxima sessão. Caso o valor de OLSR_MCTR_COUNT seja inferior ao número de caminhos existentes na tabela de rotas, representando pela variável RTABLE_SIZE, apenas parte do conjunto de rotas será utilizado, caso seja superior, todo o conjunto será utilizado. Quem controla qual caminho será utilizado é a variável PATH_COUNT. Cada caminho na lista de rotas para o destino possui uma prioridade associada. A variável PATH_COUNT define qual o caminho será utilizado segundo a sua prioridade, por exemplo, caso o valor de PATH_COUNT seja 1 a rota de maior prioridade será utilizada.

Após os valores dessas variáveis serem consultados, o OLSR MCTR executa os seguintes passos:

1. Verifica se o valor de PATH_COUNT é maior que o de OLSR_MCTR_COUNT, ou maior que o de RTABLE_SIZE;
   (a) caso o valor de PATH_COUNT seja maior que o de algum dos dois, o seu valor é alterado para 1 e o caminho com maior prioridade na lista de rotas é utilizado para transmitir o fluxo de dados;
   (b) caso o valor de PATH_COUNT seja inferior aos dois, o caminho utilizado para transmitir o fluxo de dados será aquele com a ordem de prioridade igual ao valor de PATH_COUNT.
2. Após efetuar a transmissão o valor de PATH_COUNT é incrementado;

Com esse procedimento garantimos que os pacotes serão transmitidos de forma alternada, conforme a Figura 3.2 ilustra. A seguir será descrito o funcionamento do algoritmo NIR, responsável por atualizar o valor da variável OLSR_MCTR_COUNT.

3.3.1 Cálculo do Número Ideal de Rotas (NIR)

Como visto na sessão anterior, o OLSR MCTR antes de iniciar a transmissão de um fluxo de dados RTP ele verifica o valor da variável OLSR_MCTR_COUNT. Essa variável é atualizada pelo algoritmo de cálculo do número ideal de rotas (NIR). Esse algoritmo recebe como dados de entrado o valor médio do atraso fim-a-fim da transmissão corrente.

Esse valor representa o tempo médio que os pacotes estão demorando para chegar até o destino. Sempre que o valor do atraso fim-a-fim é atualizado o algoritmo NIR é executado para também atualizar o valor da variável OLSR_MCTR_COUNT. O objetivo desse processo é fiscalizar se a transmissão de dados esta ocorrendo segundo as restrições propostas pela recomendação G 114.

Como visto no Capítulo 2, a recomendação G 114 orienta que em uma comunicação de voz em tempo real um pacote não pode demorar mais que 400 ms para chegar até o destino. Quando o atraso fim-a-fim se encontra a cima de 300 ms, o dobro do valor máximo do atraso em uma comunicação ideal que é de até 150 ms, o algoritmo NIR atualiza o valor de OLSR_MCTR_COUNT com o intuito de melhorar o desempenho da transmissão. A seguir o seu funcionamento é descrito.

1. O algoritmo NIR inicia verificando se o valor do atraso fim-a-fim é superior a 300 ms;
   (a) Caso seja, o procedimento verifica se esta sendo executa pela primeira vez;
      i. Caso esteja, o tempo corrente é armazenado e o valor de OLSR_MCTR_COUNT é mantido;
      ii. Caso não esteja, é verificado se o intervalo da última avaliação é superior a 500 ms, caso seja, o valor de OLSR_MCTR_COUNT é incrementado e o tempo corrente é armazenado, caso não, nada acontece.
2. Caso não seja, é verificado se ele se encontra entre 300 ms e 100 ms, caso se encontre, ele entende que o valor do atraso esta próximo ao valor ideal, que é de até 150 ms, e por isso apenas armazena o tempo corrente e não altera o valor de OLSR_MCTR_COUNT;
3. Caso o valor do atraso se encontre a baixo de 100 ms, é verificado se esse comportamento ocorre a mais de 10 segundos e se o valor de OLSR_MCTR_COUNT é maior que 1, caso seja, o valor de OLSR_MCTR_COUNT é decrementado e o tempo corrente armazenado, caso não seja, nada acontece;

A forma como o valor médio do atraso fim-a-fim é obtido é descrito na próxima sessão.

3.3.2 Cálculo do Atraso Fim-a-Fim

O tempo médio do atraso fim-a-fim é obtido a partir da divisão do tempo de ida e volta (*Round-Trip Time* - RTT) de um pacote na rede. O RTT é obtido a partir dos dados fornecidos pelos relatórios RTCP RR.

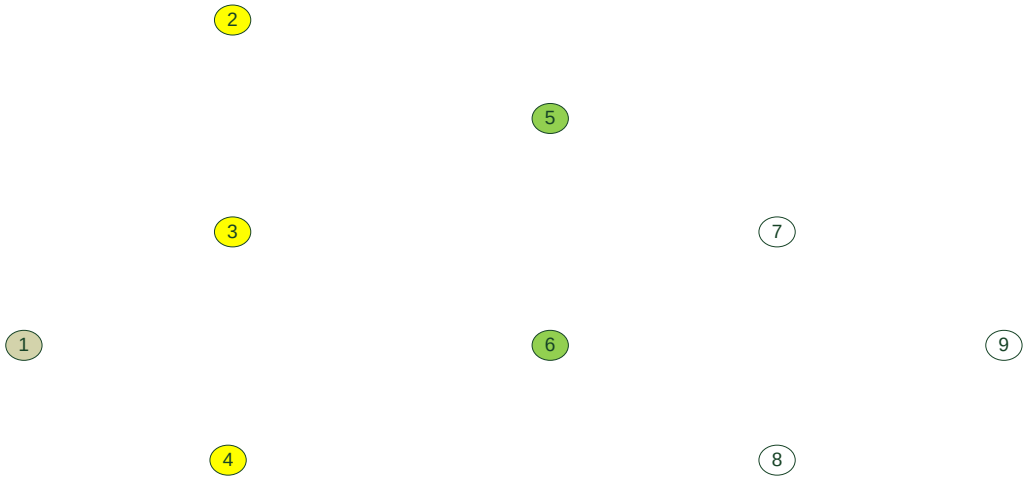

Figura 3.3: Momento 1: Nó 1 calcula sua tabela de rotas, os nós 2,3 e 4 são seus vizinhos a 1 salto e os nós 5 e 6 são seus vizinhos a 2 saltos

Quando o OLSR recebe um pacote ele o encaminha para o protocolo responsável por tratá-lo. O OLSR MCTR ao receber um pacote verifica se ele é um pacote RTCP de um receptor (RR), caso seja, antes de encaminhá-lo para o protocolo RTCP, ele busca em seu cabeçalho os campos LSR e DLSR. O campo LSR, ou *Last Sender Report*, e DLSR, ou *Delay Last Sender Report*, descritos no capítulo 2, representam consecutivamente o momento em que o nó de origem do pacote RTCP RR recebeu pela última vez um pacote RTCP *Sender Report* do nó corrente e o intervalo de tempo entra a recepção do pacote RTCP SR e o envio do RTCP RR. Com base nessas duas informações é possível calcular o tempo de ida e volta de um pacote pela rede.

O cálculo para a obtenção do RTT é simples, basta buscar o tempo corrente subtrair dele o valor presente no campo LSR e do resultado subtrair o valor do campo DLSR. Assim se obtém o valor do RTT.

3.3.3 Exemplo do Uso do Protocolo

Após apresentar o OLSR MCTR e as suas alterações em relação ao OLSR, essa sessão se dedica a exemplificar o seu funcionamento. Para isso, foi criada no NS uma cenário de teste composto por 9 nós. Todo o comportamento desse cenário foi analisado e sua descrição se encontra a seguir.

A Figura 3.3 representa o cenário de teste. O nó 1 foi tomado como referência, todos os passos serão descritos em relação ao seu comportamento. No momento inicial, após a troca de mensagens *HELLO*, o nó 1 montou as tabelas de vizinhos a 1 salto, vizinhos a 2 saltos e a tabela de topologia com o restante dos nós. A tabela 3.3.3 representa o conteúdo dessas tabelas.

Após a montagem das tabelas, o nó 1 deve calcular sua tabela de rotas e seu conjunto de MPR. A Figura 3.4 representa a configuração da tabela de rotas após executar o primeiro passo do algoritmo de cálculo de rotas descrito na sessão 3.2. A tabela3.3.3 representa a configuração

Organização das Tabelas do Nó 1 Após as Trocas de Mensagens	
Vizinhos a 1 salto	2,3,4
Vizinhos a 2 salto	5,6
Topologia	7,8,9

Tabela 3.1: Configuração padrão do meio de propagação para os cenários de teste responsáveis por comparar o desempenho do OLSR padrão com OLSR MCTR

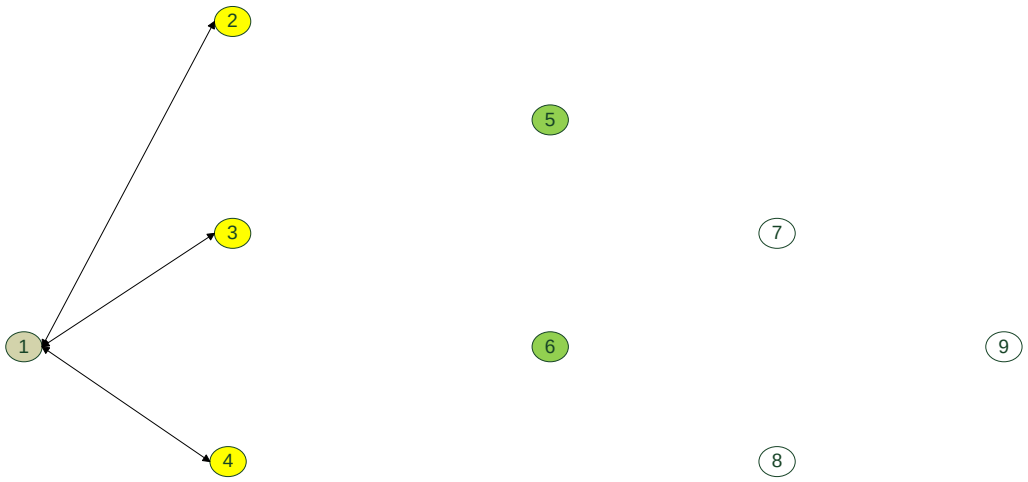

Figura 3.4: Momento 2: Nó 1 começa a montar sua tabela de rotas

da tabela de rotas após concluir o primeiro passo do algoritmo. A Figura 3.5 e a tabela 3.3.3 representam como ficou a configuração das rotas após o cálculo completo.

Configuração Inicial da Tabela de Roteamento			
R_dest_addr	R_next_addr	R_dist	R_rank
2	2	1	1
3	3	1	1
4	4	1	1

Tabela 3.2: Configuração da tabela de roteamento após montar as rotas para os destinos a 1 salto de distância

Após o nó 1 ter concluído o cálculo da tabela de rotas ele deseja enviar um fluxo de dados RTP para o nó 9. Nesse momento o OLSR MCTR consulta o valor das variáveis OLSR_MCTR_COUNT, RTABLE_SIZE e PATH_COUNT. Como o nó ainda não recebeu nenhum relatório RTCP, então o valor de OLSR_MCTR_COUNT será 1, o de RTABLE_SIZE será igual a 2 e o de PATH_COUNT será igual a 1. Dessa forma o protocolo estará transmitindo todo o fluxo apenas pela rota com maior prioridade, no caso a rota partindo do vizinho 3, Figura 3.6.

No instante seguinte, o nó 4 passa a transmitir um fluxo TCP para o nó 6. Esse fluxo congestiona o nó 6 e faz com que vários pacotes RTP sejam descartados. O nó 1, ao trocar relatórios RTPC com o nó 9, percebe a piora na qualidade da transmissão, por conta disso, o algoritmo NIR altera o valor de OLSR_MCTR_COUNT de 1 para 2, assim o segundo caminho

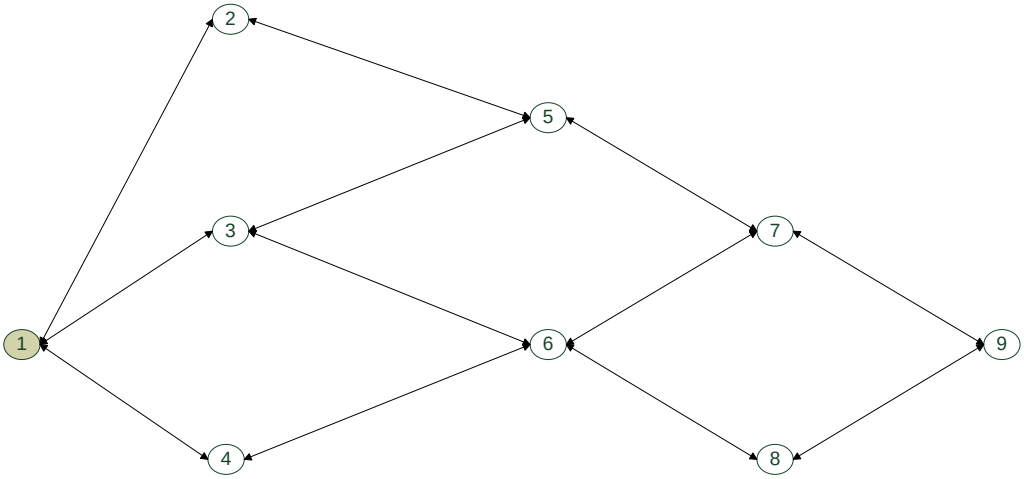

Figura 3.5: Momento 3: Calculo da tabela de rotas completo

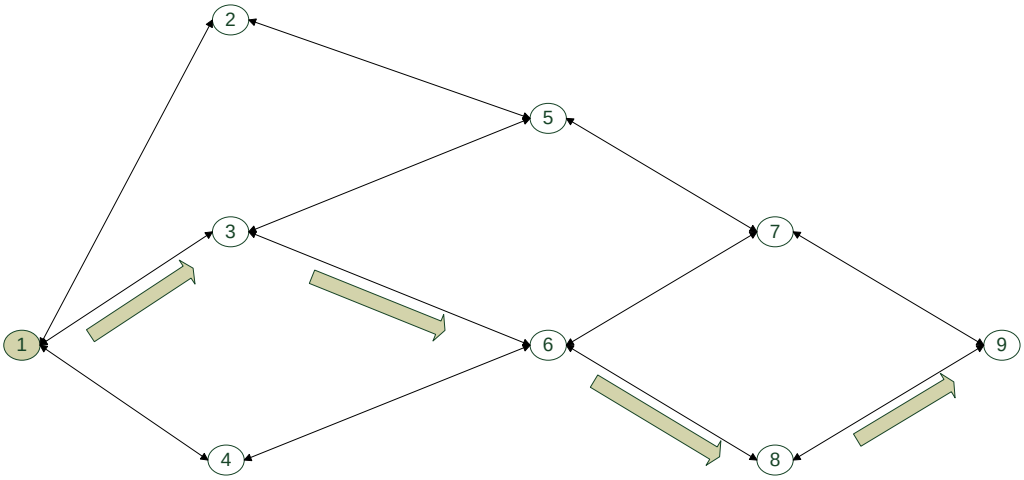

Figura 3.6: Momento 4: Nó 1 enviando dados para o nó 9 por um caminho

Configuração inicial da tabela de roteamento			
R_dest_addr	R_next_addr	R_dist	R_rank
2	2	1	1
3	3	1	1
4	4	1	1
5	2	2	1
5	3	2	2
6	3	2	1
6	4	2	2
7	2	3	1
7	3	3	2
8	3	3	1
9	3	4	1
9	2	4	2

Tabela 3.3: Tabela de roteamento após todas as rotas terem sido computadas

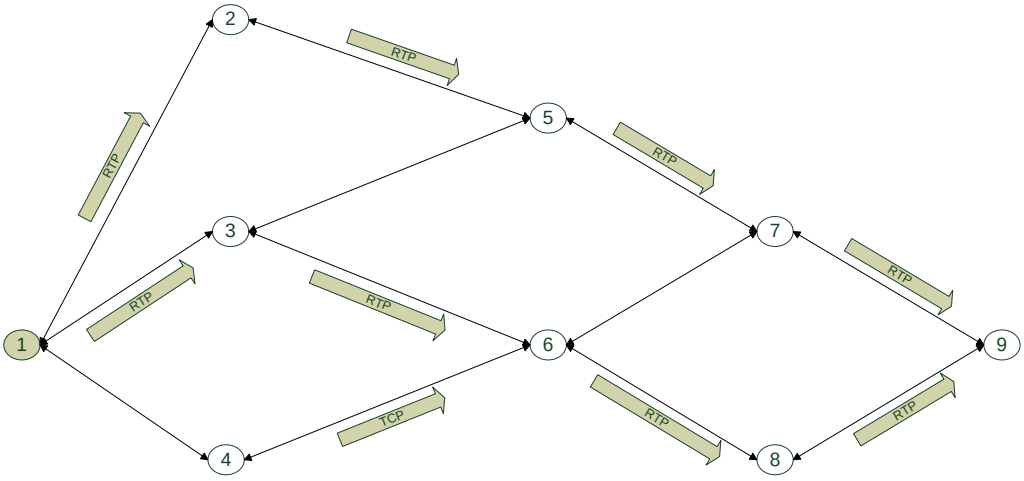

Figura 3.7: Momento 5: Nó 1 enviando dados para o nó 9 por dois caminhos

passa a ser utilizado.

Após concluir essa transmissão o nó 1 fica a espera por uma nova transferência.

4 AMBIENTE DE VALIDAÇÃO

Neste capítulo serão apresentados os cenários propostos para avaliação do OLSR MCTR. A seção 4.2 descreve o cenário de avaliação de desempenho do algoritmo NIR. A seção 4.3 descreve os cenários adotados para a comparação do desempenho do OLSR MCTR com o OLSR, AODV e AOMDV. A seção 4.4 descreve as métricas adotadas para avaliação.

4.1 Simulador

O simulador escolhido para a realização dos testes foi o *Network Simulator* 2 (NS-2), (THE...,). O NS-2 é um simulador de redes livre desenvolvido pela universidade de *Berkley* usando as linguagens C++ e TCL. Ele tem sido largamente utilizado no meio acadêmico para o desenvolvimento de várias pesquisas em redes de computadores.

Ele foi escolhido por possuir implementações estáveis para o OLSR, para o AODV, para o AOMDV e para o RTP/RTCP. A implementação do OLSR utilizada é mantida por Ros (2009). A implementação do AODV já acompanha o pacote do NS-2. A implementação do AOMDV utilizada foi desenvolvida por Das et al (2004). A implementação do RTP/RTCP pertence ao pacote padrão do NS-2 e foi recentemente atualizada por Bouras et al (2008).

4.2 Cenário de Validação do Algoritmo de Cálculo do Número Ideal de Rotas (NIR)

O algoritmo de cálculo do número ideal de rotas (NIR) é parte fundamental desse trabalho. Para determinar se sua escolha foi apropriada, foi desenvolvido um cenário controlado para testes.

O cenário é composto por 20 nós postados segundo a topologia definida na Figura 4.1. O meio de propagação segue a configuração informada na tabela 4.1.

Devido à necessidade de se obter um ambiente controlado para avaliar apropriadamente o comportamento do algoritmo, todos os nós são estáticos. Nesse cenário, o RTP tenta realizar uma transmissão de boa qualidade entre os nós 0 e 7. Existem dois caminhos possíveis para o fluxo de dados a seguir. O primeiro deles vai do nó 1 até o nó 6 e o segundo vai do nó 10 até o 19. O OLSR original tende a enviar o fluxo de dados pelo caminho com o menor número

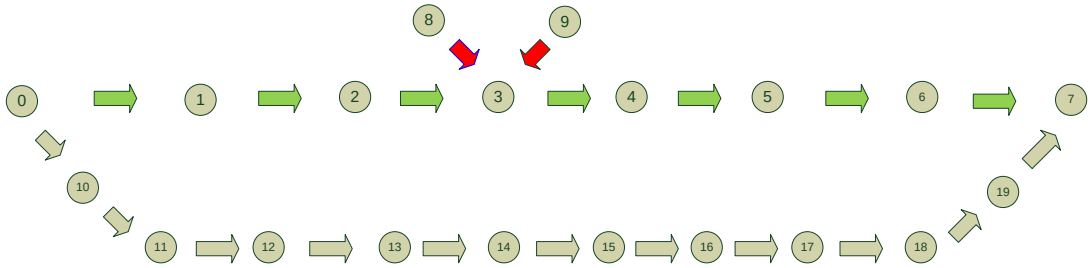

Figura 4.1: Cenário de teste do algoritmo NIR

Validação do Algoritmo de Cálculo de Rotas	
Dimensões	2000m x 2000m
Tempo de simulação	100s
Taxa de transmissão de dados	Constant Bitrate (CBR)
Camada MAC	802_11Ext
Camada Física	WirelessPhyExt
Frequência	2,4 GHz
Modelo de propagação	TwoRayGround

Tabela 4.1: Configuração do meio de propagação para o cenário de teste do algoritmo NIR

de saltos. Neste caso o fluxo sai do nó 0 para o nó 1 e segue pelos nós de número 2 a 6 até finalmente alcançar o nó 7.

Para validar o algoritmo proposto, será necessário comprometer a capacidade de processamento de pacotes de um dos nós no caminho. Ao longo da simulação os nós 8 e 9 transmitem um fluxo TCP (ROSE; CASS, 1987) para o nó 3.A sequência de transmissões é descrita na tabela 4.2. Dessa forma o nó 3 fica ocupado com os pacotes endereçados a ele e não consegue encaminhar adequadamente os outros pacotes.

Validação do Algoritmo de Cálculo de Rotas	
0 até 100	Início do fluxo de dados RTP do nó 0 até o nó 7
15 até 30	Inclusão do fluxo de dados TCP dos nós 8 e 9 para o nó 3
50 até 65	Inclusão do fluxo de dados TCP dos nós 8 e 9 para o nó 3

Tabela 4.2: Comportamento dos nós no cenário de testes responsável por validar o algoritmo NIR

Entre os instantes 0 e 14 apenas o nó 0 está transmitindo dados para o nó 7. No instante 15, os nós 8 e 9 iniciam a transmissão de um tráfego TCP para o nó 3. O nó 3 passará a processar mais lentamente o fluxo RTP chegando, em alguns instantes, a descartar alguns pacotes. Assim o menor caminho passa a ter um gargalo em sua composição. Caso o nó 0, fazendo uso das informações contidas nos relatórios RTCP, perceba um aumento no atraso fim-a-fim nos instantes onde o nó 3 esta sobrecarregado, espera-se o uso do caminho alternativo em conjunto com o caminho mais curto para tentar diminuir esse atraso.

No total foram realizadas 33 simulações utilizando esse cenário, cada uma com duração

de 100 segundos. De cada cenário foram coletados os valores para o atraso fim-a-fim e o número de rotas efetivamente utilizadas em intervalos de tempo definidos na tabela 4.3. Esses intervalos foram definidos segundo o comportamento do trafego de fundo proposto e do funcionamento do algoritmo NIR. Em seguida foi retirado o valor médio para cada métrica em cada um dos intervalos e os resultados foram analisados. O objetivo dessa análise é verificar se algoritmo NIR, ao perceber a degradação da qualidade da transmissão, irá alterar o número de caminhos utilizados e se isso ajudará a transmissão a alcançar valores apropriados de atraso fim-a-fim.

Tempo de Parada para Análise	
1	Início da simulação
14, 15, 16	Início do tráfego de fundo TCP
24, 25, 26	10 segundos do início do tráfego de fundo
30, 31	O tráfego de fundo acaba
35	10 segundos da última checagem do algoritmo NIR
49, 50, 51	Início do tráfego de fundo TCP
59, 60, 61	10 segundos do início do tráfego de fundo
69, 70	10 segundos da última checagem do algoritmo NIR
98	Verificar a atualização do número de caminhos

Tabela 4.3: Tempos de parada do

4.3 Cenários Para Avaliação da Performance dos Protocolos

Para avaliar a performance dos 4 protocolos foram gerados 5 cenários a partir daqueles propostos no trabalho de Oo et al. (2010). Em seu trabalho, Oo et al. (2010), comparou o desempenho do OLSR com o AOMDV. As métricas utilizadas foram o atraso fim-a-fim, a percentagem de dados entregues e a carga normalizada de roteamento, sendo as duas primeiras também utilizadas para a avaliação do OLSR MCTR.

A configuração dos cenários se encontra descrita na tabela 4.3. Os cenários foram gerados a partir do programa BonnMotion (ASCHENBRUCK et al., 2010), o mesmo utilizado no trabalho de Oo et al. (2010) para gerar os seus cenários. O tráfego de fundo existente na rede se encontra descrito na tabela 4.5. Cada cenário foi executado 33 vezes

4.4 Métricas

As métricas escolhidas para avaliar os protocolos são:

1. atraso fim-a-fim;
2. variação do atraso (*Jitter*)
3. relação entre *bytes* recebidos e *bytes* enviados.

Validação do Algoritmo de Cálculo de Rotas	
Dimensões	1000m x 1000m
Tempo de simulação	200s
Taxa de transmissão de dados	Constant Bitrate (CBR)
Camada MAC	802_11Ext
Camada Física	WirelessPhyExt
Frequência	2,4 GHz
Modelo de propagação	TwoRayGround
Número de nós	50
velocidades	5, 10, 12, 15, 20
Alcance dos nós	250

Tabela 4.4: Configuração padrão do meio de propagação para os cenários de teste responsáveis por comparar o desempenho do OLSR padrão com OLSR MCTR

Tráfego de Fundo dos Cenários	
10 até 60	Fluxo de dados TCP dos nós 8 até o nó 10 e do nó 24 até o 16
30 até 80	Fluxo de dados TCP dos nós 54 até o nó 2 e do nó 20 até o 5
50 até 100	Fluxo de dados TCP dos nós 40 até o nó 31 e do nó 23 até o 51
70 até 120	Fluxo de dados TCP dos nós 32 até o nó 19 e do nó 59 até o 12
90 até 140	Fluxo de dados TCP dos nós 21 até o nó 35 e do nó 24 até o 16
110 até 160	Fluxo de dados TCP dos nós 37 até o nó 32 e do nó 48 até o 6
130 até 190	Fluxo de dados TCP dos nós 53 até o nó 41 e do nó 43 até o 18

Tabela 4.5: Tráfego de fundo presente na rede

A primeira métrica a ser observada é o atraso fim-a-fim. Segundo Júnior (2008), atraso é a medida do tempo decorrido entre o envio de uma mensagem por um nodo e a recepção desta mensagem pelo nodo destino. Isto ocorre devido aos possíveis atrasos causados pela latência da descoberta de rotas, velocidade de transmissão do meio, congestionamentos, por causa do processamento ou espera na fila de um dispositivo intermediário.

A segundo métrica analisada foi a variação do atraso ou *Jitter*. O *Jitter* pode ser entendido como a variação no tempo e na sequência de entrega de pacotes, devido à variação do atraso (MARTINS, 2008). Segundo MELLO (2001), jitter é uma distorção que acontece, por exemplo, quando fluxos de voz ou vídeo são transmitidos em uma rede e os pacotes não chegam ao seu destino dentro da ordem sucessiva ou em uma determinada cadência, ou seja, eles variam em termos de tempo de atraso. Essa distorção é particularmente prejudicial a tráfegos de dados multimídia, fazendo com que o sinal de áudio ou vídeo tenha uma qualidade distorcida ou fragmentada na recepção. Uma técnica comumente utilizada para minimizar a variação de atraso é a utilização de buffer, o qual vai armazenando os dados a medida que eles chegam e os encaminham para a aplicação a uma mesma cadência.

A terceira métrica é a percentagem de *bytes* efetivamente entregues é obtida a partir da relação entre o número de *bytes* recebidos pelo nó destino e o número de *bytes* enviados pelo nó emissor. Ela representa a eficiência de transmissão do protocolo. É possível existir protocolos

com um alto atraso fim-a-fim e uma boa taxa de transmissões efetuadas com sucesso.

5 RESULTADOS DE SIMULAÇÃO

Neste capítulo serão apresentados os resultados das simulações. A seção 5.1 apresenta o comportamento do OLSR MCTR em função do algoritmo NIR. A seção 5.2 analisa o desempenho dos protocolos OLSR, AODV, OLSR MCTR e AOMDV nos cenários propostos no capítulo anterior. As métricas adotadas foram atraso fim-a-fim, relação entre *bytes* recebidos e *bytes* enviados.

5.1 Avaliação de Desempenho do Algoritmo de Cálculo do Número Ideal de Rotas (NIR)

Para realização desse teste foi desenvolvido um cenário com 20 nós estáticos. No total foram realizadas 33 simulações com duração de 100 segundos cada. Foram retiradas as médias para as métrica atraso fim-a-fim e número de caminhos realmente utilizados em intervalos de tempo definidos na tabela 4.3.

Os gráficos 5.1 e 5.2 apresentam o comportamento do atraso fim-a-fim e o número ideal de caminhos indicados ao longo do tempo.

Ao observarmos as Figuras 5.1 e 5.2 no instante 1 percebemos que a transmissão ocorre com um atraso baixo e é utilizado apenas um caminho. No instante 15 tem início as transmissões de dados do nó 8 e 9 para o nó 3. Analisando o intervalo de tempo que vai do instante 14 até o 16, percebemos um pico no atraso fim-a-fim no instante 15 e o seu retorno ao patamar anterior no momento seguinte. Também ocorre a elevação no número de caminhos utilizados, de 1 pra 2 no instante 16, dessa forma é possível dizer que o uso do segundo caminho permite ao protocolo transmitir com maior eficiência o fluxo de dados.

No intervalo de tempo que vai dos instantes 24 a 26 percebemos no instante 25 que o protocolo tem utilizar apenas um caminho e esse movimento é acompanhado por um pico no gráfico do atraso fim-a-fim, em seguida o protocolo volta a utilizar dois caminhos e o atraso retorna para valores a baixo de 300 ms. Esse comportamento era esperado, pois o algoritmo NIR apos 10s da transmitindo com atraso a baixo de 100 ms, tenta realizar a transmissão por um número menor de caminhos.

No intervalo de tempo que vai dos instantes 30 a 31, a transmissão dos nós 8 e 9 para, porém o protocolo continua a transmitir por dois caminhos. Esse comportamento é esperado, pois o algoritmo NIR só reduz o número de caminhos apos 10s de transmissão com atraso a

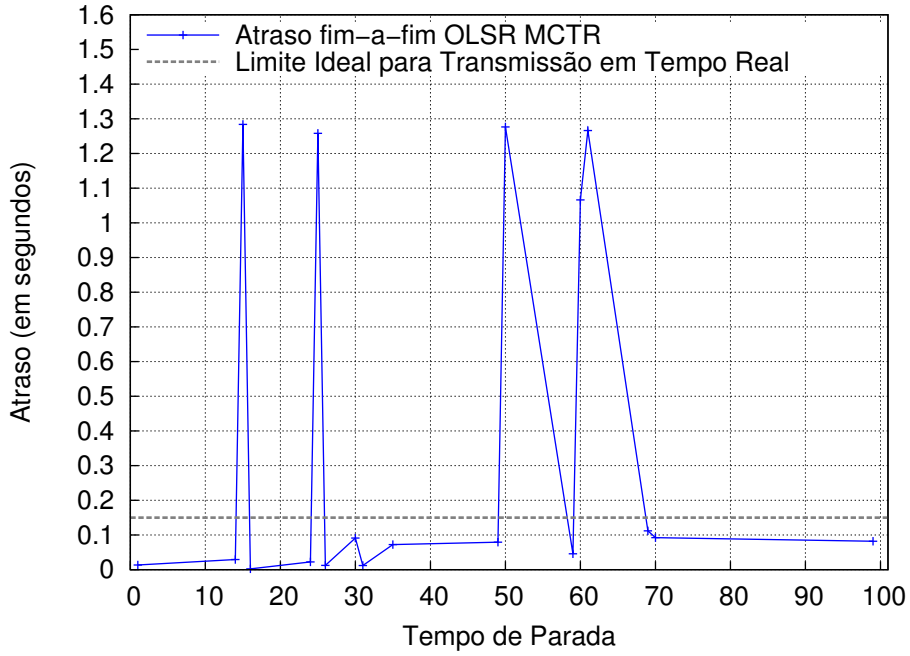

Figura 5.1: Resultado do atrazo fim-a-fim em uma das iterações

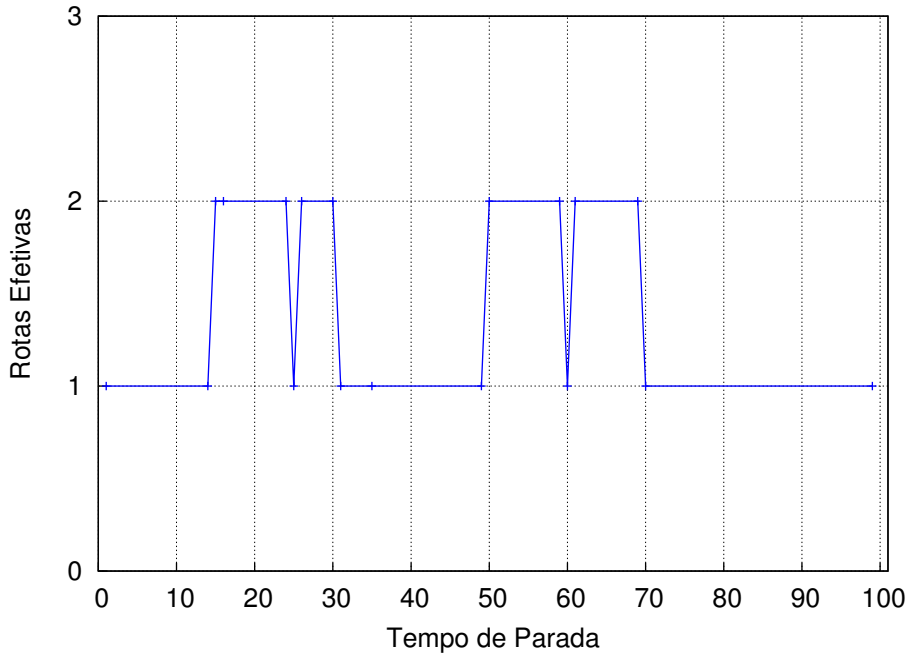

Figura 5.2: Número ideal de caminhos representado pela variável OLSR_MCTR_COUNT em uma das iterações

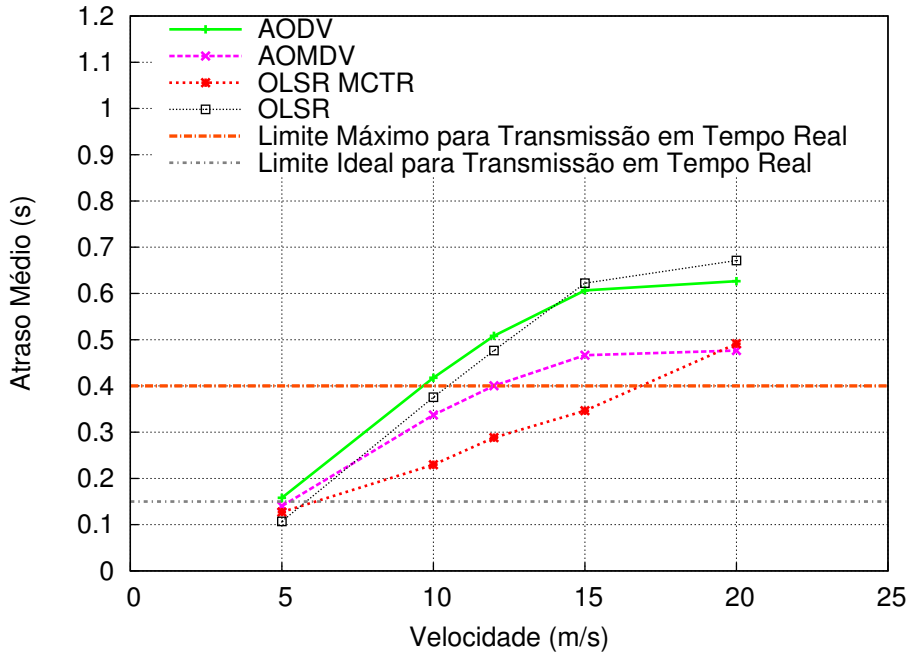

Figura 5.3: Atraso fim-a-fim médio de cada simulação

baixo de 100 ms, o que ocorreu no instante 35.

No instante 50 os nós 8 e 9 voltam a transmitir para o nó 3, o que ocasiona um pico no atraso fim-a-fim no instante 50. Devido a isso, o protocolo volta a transmitir por 2 caminhos no instante 51, transmitindo com maior eficiência o fluxo de dados e consequentemente diminui o valor do atraso médio.

No intervalo de tempo que vai de 59 a 61, ocorre o mesmo fenômeno observado no intervalo de 24 a 26, o protocolo volta utilizar apenas um caminho, o atraso médio sobe e em seguida o número de caminhos utilizados volta a ser 2. No instante 65 os nós 8 e 9 param suas transmissões. No instante 70 o protocolo volta a transmitir por meio de um caminho apenas. Uma última verificação é realizada no instante 98 onde observamos a transmissão sendo executada com o atraso a baixo de 300 ms por meio de um único caminho.

Com esses resultados podemos confirmar a validade do Algoritmo proposto. O próximo passo é a validação do protocolo.

5.2 Avaliação de Desempenho do OLSR MCTR

5.2.1 Atraso Fim-a-Fim

A figura 5.3 apresenta o gráfico com os resultados do atraso fim-a-fim para cada um dos protocolos nos cenários propostos. No cenário 1, nós se movimentam a velocidade de 5 m/s, podemos observar o bom comportamento dos quatro protocolos. Nesse cenário o OLSR apresentou o melhor resultado para essa métrica. Isso se deve ao fato da rede estar em excelentes

condições para o seu funcionamento. Os nós se movimentam a uma velocidade relativamente baixa e uma boa quantidade de nós intermediários se encontra entre o nó origem e o nó destino.

O OLSR MCTR obteve um resultado próximo ao do OLSR. Ele fez uso do mesmo número de caminhos que ele. O principal motivo para a diferença de resultados entre eles foi o processamento extra existente no primeiro. O OLSR MCTR tem de analisar os dados dos pacotes RTCP RR e definir o número ideal de rotas, passos não executados pelo OLSR.

O AOMDV, por ser um protocolo reativo, obteve resultados piores aos apresentados pelos dois protocolos anteriores, o seu desempenho foi superior apenas ao observado pelo AODV. Todas as médias estão dentro do intervalo entre 100 ms e 170 ms.

No cenário 2 os nós estão se deslocando a uma velocidade de 10 m/s. O desempenho de todos os protocolos nesses foi inferior ao observado no cenário anterior. Isso se deve ao fato de quanto mais rápido à rede se modifica menor é o tempo de existência das rotas. Assim todos os protocolos necessitam atualizar suas tabelas de rotas com a constância maior. Esse passo é custoso, pois enquanto esse cálculo não for concluído os pacotes não tem como efetuar suas transmissões.

Analisando esse cenário é possível observar um melhor comportamento dos protocolos capazes de fazer uso de múltiplos caminhos. O OLSR MCTR e o AOMDV sofreram tanto quanto os outros dois protocolos, pois não necessitam atualizar suas tabelas de rotas com a mesma frequência do OLSR e do AODV.

O OLSR MCTR foi superior ao AOMDV, pois faz uso das informações presentes nos relatórios RTCP RR. Esses dados orientam o OLSR MCTR a respeito de quando as rotas que ele esta utilizando ainda são apropriadas para a transmissão de dados em tempo real e quando o cálculo de novas rotas é necessário. Uma rota pode deixar de ser viável quando um enlace é rompido, ou quando alguns dos nós que formam a rota estão sobrecarregados.

No cenário 3 os nós passam a se movimentar a 12 m/s, um pequeno aumento de velocidade em comparação com o cenário anterior. Essa mudança foi o suficiente para as métricas de todos os protocolos piorarem, o OLSR MCTR, pelo mesmo motivo apresentado anteriormente, teve uma menor degradação em suas métricas em comparação ao restante dos protocolos. Apenas ele se manteve completamente dentro do intervalo considerado aceitável pela recomendação G 114. O AOMDV obteve valores em torno do limite aceitável. Os outros protocolos estão à cima do limite considerado aceitável.

No cenário 4 a velocidade dos nós passa a ser de 15 m/s. Novamente com o aumento da velocidade foi verificado a piora nas métricas de todos os protocolos. Tirando o OLSR MCTR todos estão a cima do limite estabelecido pela recomendação G 114.

No cenário 5 a velocidade dos nós passa a ser de 20 m/s. Nesse ambiente o AOMDV passou a ter o melhor desempenho em comparação ao restante dos protocolos seguido pelo OLSR MCTR. Isso se deve, pois o OLSR MCTR atualiza mais vezes suas rotas em comparação com os cenários anteriores. Todos os protocolos estão acima da meta de 400 ms.

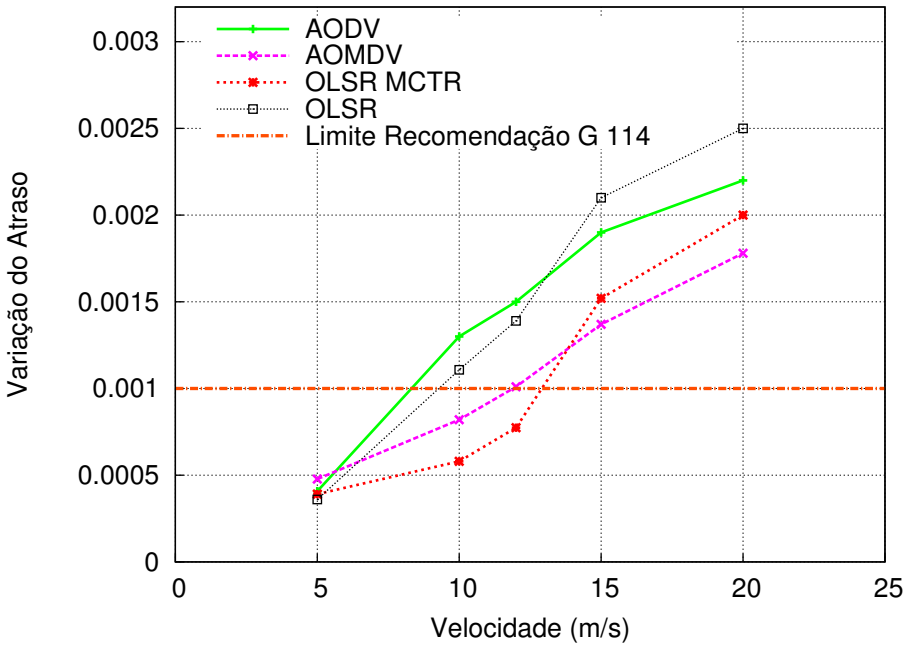

Figura 5.4: Variação do atraso em cada simulação

5.2.2 Variação do Atraso

A figura5.4 representa o comportamento da variação do atraso de cada protocolo ao longo de cada simulação. Analisando as informações do gráfico observamos o aumento da variação a medida que a velocidade de deslocamento dos nós aumenta.

A figura 5.5 representa o uso médio de caminhos pelo AOMDV e pelo OLSR MCTR a medida que a velocidade dos nós vai aumentando. Comparando os dois gráficos verificamos que o aumento no uso de múltiplos caminhos acompanha o aumento da variação do atraso. O mesmo ocorre com o AOMDV porem não com a mesma intensidade.

A transmissão paralela de dados, feita utilizando múltiplos caminhos, interfere na ordem com que os pacotes chegam até o destino. Como cada caminho é percorrido com uma velocidade diferente, os pacotes tendem a chegar desordenados no destino. Isso explica o porquê da variação do atraso no OLSR MCTR crescer mais em comparação com o AOMDV, o OLSR MCTR utilizou nos dois últimos cenários mais rotas que o AOMDV.

Para se manter uma transmissão em tempo real o intervalo entre a chegada de cada pacote deve ser de no máximo 1ms, segundo a recomendação G 114, a cima disso a reprodução do conteúdo é comprometida. Assim, no cenário 4 o OLSR MCTR, mesmo entregando os pacotes dentro do intervalo de atraso desejado, esta tendo pacotes descartados pelo *playout buffer* e assim comprometendo a qualidade da transmissão.

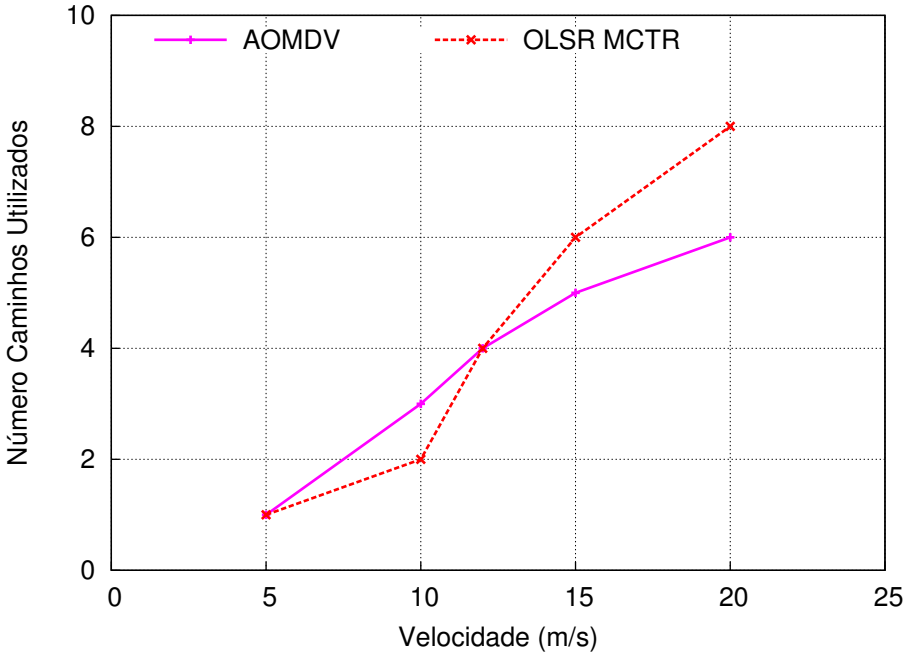

Figura 5.5: Uso médio de caminhos pelo OLSR MCTR e AOMDV

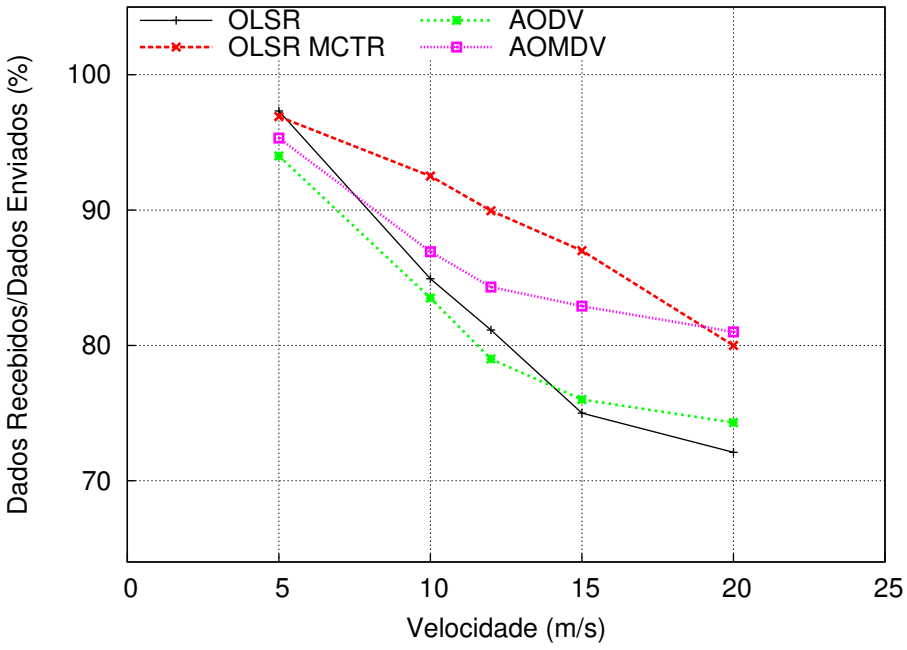

Figura 5.6: Percentagem de dados recebidos em função dos enviados em cada simulação

5.2.3 Percetagem de Dados Entregues

A figura 5.6 representa a percentagem de dados efetivamente entregues em cada um dos cenários. Podemos verificar a diminuição da taxa de entrega acompanhar o aumento no atraso fim-a-fim. Devido às alterações na velocidade dos nós e o rompimento mais rápido dos enlaces. Nessas condições o OLSR MCTR e o AOMDV obtiveram os melhores resultados.

5.2.4 Trabalhos Relacionados

O trabalho de Lu et al. (2008) definiu um protocolo baseado no AOMDV para transmissão de vídeos em tempo real chamado de *Link Stability AOMDV* (LS-AOMDV). O LS-AOMDV seleciona os caminhos de acordo com a estabilidade do primeiro enlace. Além disso, é definido um novo mecanismo de manutenção de rotas capaz de reduzir o impacto da mobilidade dos nós no transporte do fluxo de dados. Nas simulações executadas, o LS-AOMDV apresenta desemprenho superior ao do AODV e ao AOMDV.

O trabalho de Gowrishankar et al. (2009) compara o desempenho dos quatro protocolos de roteamento AODV, AODVUU, AOMDV e RAODV sob vários cenários de rede. Em todos os cenários o AOMDV e o RAODV apresentam um comportamento superior ao restante dos protocolos.

O trabalho de Oo et al. (2010) examinou o comportamento do OLSR e do AOMDV utilizando TCP-Reno (FLOYD; HENDERSON; GURTOV, 2004). Os resultados indicaram um melhor desempenho geral do OLSR em redes onde o tráfego é mais esparso, porém em redes escassas de recursos, o seu desempenho tende a cair. O AOMDV é indicado para redes com alta mobilidade dos nós.

O trabalho de Gowrishankar t.g. basavaraju (2007) compara o desempenho dos protocolos OLSR e AODV. O AODV apresentou melhor desempenho em ambientes com poucos recursos, enquanto o OLSR se apresenta em redes mais densas e com menor mobilidade dos nós.

5.2.4.1 Análise Final

Tal qual o LS-AOMDV, o OLSR MCTR leva em consideração o estado dos enlaces para o cálculo de rotas e faz uso de um mecanismo para melhorar a capacidade de adaptação às alterações da rede. Esse mecanismo faz uso dos relatórios RTCP para definir a quantidade de múltiplas rotas utilizáveis no processo de encaminhamento dos fluxos de dados. Assim, o OLSR MCTR demonstrou ser uma boa alternativa para a transferência de dados em tempo real, tal o qual o LS-AOMDV também o demonstrou.

Os trabalhos de Mao et al. (2006), Oo et al. (2010) e Gowrishankar t.g. basavaraju (2007), realizaram comparações entre diversos protocolos, entre eles o OLSR, AOMDV, AODV. O trabalho de Mao et al. (2006) demonstrou como o AOMDV é superior ao AODV. Os resultados apresentados estão de acordo com essa afirmação, pois nos 5 cenários avaliados o AOMDV

apresentou desempenho superior em todos eles.

Os trabalhos de Oo et al. (2010) e Gowrishankar t.g. basavaraju (2007) comparam ou o AOMDV, ou o AODV, ao OLSR. Os dois chegam a conclusões equivalentes: o OLSR apresentou um melhor desempenho em redes mais densas, enquanto os outros dois apresentam um melhor desempenho em redes menos densas e de alta mobilidade. Essas afirmações estão de acordo com os resultados apresentados.

O OLSR MCTR apresenta um desemprenho inferior ao do AOMDV quando os nós passaram a se movimentar a 20 m/s. Nesse ambiente os enlaces passaram a se romper mais rapidamente isso provocou o aumento do número de vezes em que o procedimento de atualização de rotas é chamado no OLSR MCTR. O fato do AOMDV apresentar um melhor desempenho nessas condições esta de acordo com os trabalhos apresentados.

6 CONCLUSÃO

Neste capítulo são abordadas as contribuições deste trabalho no subitem 6.1, e as sugestões para trabalhos futuros no 6.2.

6.1 Contribuições

Conforme enunciado na seção 1.2, o objetivo desse trabalho é propor um protocolo de roteamento para rede Ad Hoc capaz de suportar transmissões de dados em tempo real, esse protocolo foi denominado OLSR MCTR. Ele faz uso de uma estratégia *cross layer*, ou seja, utiliza elementos de camadas diferentes para melhorar a transmissão de dados em tempo real em redes Ad Hoc. O OLSR MCTR faz uso de dados contidos nos relatório RTCP para definir a quantidade de caminhos utilizados para a transferência de dados e quando é a melhorar hora para atualizar sua tabela de rotas.

A avaliação desse trabalho foi realizada em ambiente de simulação. O OLSR MCTR foi implementado no simulador de redes *Network Simulator 2* (NS-2) a partir da implementação do OLSR de Ros (2009). O seu desempenho foi comparado com o do OLSR, com o do AODV e com o do AOMDV implementado por Das et al (2004). A implementação utilizada para o RTP/RTCP foi a de Bouras et al. (2008).

Esse quatro protocolos foram avaliados em 5 cenários diferentes. Para cada cenário foram realizadas 33 simulações. De cada simulação foram retiradas as seguintes métricas:

1. média do atraso fim-a-fim;
2. média da relação entre *bytes* recebidos e *bytes* enviados.

Foi avaliado o comportamento das médias de cada simulação para cada cenário. Os resultados mostram o OLSR MCTR tendo um resultado próximo ao OLSR, AODV e AOMDV em ambientes onde os nós se deslocam a uma velocidade média baixa. A medida que a velocidade de deslocamento dos nós aumenta, o tempo de vida das rotas diminui. Com isso existe a necessidade de aumentar a frequência de atualização das tabelas de rotas de cada protocolo. Esse procedimento ocasiona um maior atraso na transmissão dos dados. Os protocolos com os melhores desempenhos foram aqueles que dispunham de opções de caminhos alternativos para transmitir os seus pacotes, AOMDV e OLSR MCTR. Dentre esse o OLSR MCTR obteve

o melhor desempenho, pois, devido ao uso das informações dos relatórios RTCP RR, foi capaz de definir não somente quantos caminhos utilizar e também qual a hora certa de atualizar sua tabela de rotas. Um dado interessante foi o melhor comportamento do AOMDV no último cenário, mostrando a possibilidade de uso desse protocolo em ambientes de baixa densidade e alto deslocamento dos nós.

6.2 Trabalhos Futuros

O OLSR MCTR ainda não teve todo o seu desempenho avaliado. Novos trabalhos devem ser executados para analisar qual deve ser o seu melhor uso com pacotes diferentes do RTP e como podemos melhorar o mecanismo de cálculo de rotas para redes onde os nós se movimentem a velocidades maiores. Uma proposta seria utilizar conceitos de cinemática no processo de cálculo de rotas e assim obter rotas que existam durante um período de tempo maior.

REFERÊNCIAS BIBLIOGRÁFICAS

AL, A. A. E.; SAADAWI, T.; LEE, M. Unequal error protection for real-time video in mobile ad hoc networks via multi-path transport. *Comput. Commun.*, Butterworth-Heinemann, Newton, MA, USA, v. 30, n. 17, p. 3293–3306, 2007. ISSN 0140-3664.

ALEXANDER, B. *802.11 Wireless Network Site Surveying and Installation (Networking Technology)*. [S.l.]: Cisco Press, 2004. ISBN 1587051648.

ALTURKI, R.; MEHMOOD, R. Multimedia ad hoc networks: Performance analysis. IEEE Computer Society, Washington, DC, USA, p. 561–566, 2008.

ASCHENBRUCK, N. et al. Bonnmotion: a mobility scenario generation and analysis tool. In: *SIMUTools '10: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010. p. 1–10. ISBN 78-963-9799-87-5.

BOUKERCHE, A. Performance evaluation of routing protocols for ad hoc wireless networks. *Mob. Netw. Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 9, p. 333–342, August 2004. ISSN 1383-469X. Disponível em: <http://dx.doi.org/10.1145/1012215.1012224>.

BOURAS, C. *A Framework for Cross Layer Adaptation for Multimedia Transmission over Wired and Wireless Networks*.

BOURAS, C.; GKAMAS, A.; KIOUMOURTZIS, G. *Performance Evaluation of Cross Layer Adaptive Multimedia Transmission: The case of Wired Networks*.

CLAUSEN, T.; JACQUET, P. *Optimized Link State Routing Protocol (OLSR)*. IETF, out. 2003. RFC 3626 (Experimental). (Request for Comments, 3626). Disponível em: <http://www.ietf.org/rfc/rfc3626.txt>.

CONTI, M.; GREGORI, E.; TURI, G. A cross-layer optimization of gnutella for mobile ad hoc networks. In: *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2005. p. 343–354. ISBN 1-59593-004-3.

DAS, S.; MARINA, M. Aomdv. In: CMU/MONARCH GROUP CARNEGIE MELLON UNIVERSITY. [S.l.], 2004.

DAY, J. The (un)revised osi reference model. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 25, n. 5, p. 39–55, 1995. ISSN 0146-4833.

FLOYD, S.; HENDERSON, T.; GURTOV, A. *The NewReno Modification to TCP's Fast Recovery Algorithm*. IETF, abr. 2004. RFC 3782 (Proposed Standard). (Request for Comments, 3782). Disponível em: <http://www.ietf.org/rfc/rfc3782.txt>.

GOWRISHANKAR, S.; SARKAR, S.; BASAVARAJU, T. Performance analysis of aodv, aodvuu, aomdv and raodv over ieee 802.15.4 in wireless sensor networks. In: . [S.l.: s.n.], 2009. p. 59 –63.

GOWRISHANKAR T.G. BASAVARAJU, M. S. S. K. S. S. Scenario based performance analysis of aodv and olsr in mobile ad hoc network. *Proceedings of the 24th South East Asia Regional Computer Conference*, 2007.

GROUP, A.-V. T. W. et al. *RTP: A Transport Protocol for Real-Time Applications*. IETF, jan. 1996. RFC 1889 (Proposed Standard). (Request for Comments, 1889). Obsoleted by RFC 3550. Disponível em: <http://www.ietf.org/rfc/rfc1889.txt>.

KUBINIDZE, N.; GANCHEV, I.; O'DROMA, M. Network Simulator NS2: Shortcomings, Potential Development and Enhancement Strategies. *Modelling and Simulation Tools for Emerging Telecommunications Networks*, Springer, v. 1, p. 263–277, 2004.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. 5th. ed. USA: Addison-Wesley Publishing Company, 2009. ISBN 0136079679, 9780136079675.

LI, T. et al. Topology mismatch avoidable cross-layer protocol for p2p file discovery in manets. In: *WCNC'09: Proceedings of the 2009 IEEE conference on Wireless Communications & Networking Conference*. Piscataway, NJ, USA: IEEE Press, 2009. p. 2943–2947. ISBN 978-1-4244-2947-9.

LIN, C.-h. et al. Supporting real-time speech on wireless ad hoc networks: inter-packet redundancy, path diversity, and multiple description coding. In: *WMASH '04: Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*. New York, NY, USA: ACM, 2004. p. 11–20. ISBN 1-58113-877-6.

LIU, G. H. Z.; LIU, Z. An adaptive control scheme for real-time mpeg4 video over ad-hoc networks. *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, IEEE Computer Society, Washington, DC, USA, v. 2, p. 1153–1155, 2005. ISSN 0-7803-9335-X.

LU, Q. et al. Improved multi-path aodv protocols for real-time video transport over mobile ad hoc networks. In: . [S.l.: s.n.], 2008. v. 1, p. 621 –625.

MAO, S. et al. Mrtp: a multiflow real-time transport protocol for ad hoc networks. *IEEE Transactions on Multimedia*, v. 8, n. 2, p. 356–369, 2006.

MAO, S. et al. Mrtp: a multiflow real-time transport protocol for ad hoc networks. *Multimedia, IEEE Transactions on*, v. 8, n. 2, p. 356 – 369, apr. 2006. ISSN 1520-9210.

MAO, S. et al. Multipath video transport over ad hoc networks. *IEEE Wireless Communications*, v. 12, n. 4, p. 42–49, 2005.

MAO SHUNAN LIN, P. S. Y. W. C. E. S. Video transport over ad hoc networks: multistream coding with multipath transport. *Selected Areas in Communications, IEEE Journal on*, IEEE Computer Society, Washington, DC, USA, v. 21, p. 1721–1737, 2003. ISSN 0733-8716.

MARCONDES, C. A. C. et al. Ambiente para simulação e monitoração de ligações telefônicas ip. *XXI Simpósio Brasileiro de Redes de Computadores*, SBC, 2003.

MARTINS, J. Qualidade de serviço (qos) em redes ip, principios básicos, parâmetros e mecanismos. In: . [S.l.: s.n.], 2008.

NGUYEN, N. T. et al. Electric-field-based routing: a reliable framework for routing in manets. *SIGMOBILE Mob. Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 8, n. 2, p. 35–49, 2004. ISSN 1559-1662.

OO, M.; OTHMAN, M. Performance comparisons of aomdv and olsr routing protocols for mobile ad hoc network. In: . [S.l.: s.n.], 2010. v. 1, p. 129 –133.

PERKINS, C.; BELDING-ROYER, E.; DAS, S. *Ad hoc On-Demand Distance Vector (AODV) Routing*. IETF, jul. 2003. RFC 3561 (Experimental). (Request for Comments, 3561). Disponível em: <http://www.ietf.org/rfc/rfc3561.txt>.

QIN, M.; ZIMMERMANN, R. Improving mobile ad-hoc streaming performance through adaptive layer selection with scalable video coding. In: LIENHART, R. et al. (Ed.). *ACM Multimedia*. [S.l.]: ACM, 2007. p. 717–726. ISBN 978-1-59593-702-5.

ROS, F. J. Um-olsr. In: MANET SIMULATION AND IMPLEMENTATION AT THE UNIVERSITY OF MURCIA. [S.l.], 2009.

ROSE, M.; CASS, D. *ISO Transport Service on top of the TCP Version: 3*. IETF, maio 1987. RFC 1006 (Standard). (Request for Comments, 1006). Updated by RFC 2126. Disponível em: <http://www.ietf.org/rfc/rfc1006.txt>.

SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. IETF, jul. 2003. RFC 3550 (Standard). (Request for Comments, 3550). Updated by RFC 5506. Disponível em: <http://www.ietf.org/rfc/rfc3550.txt>.

THE network simulator - ns-2. Disponível em: <http://www.isi.edu/nsnam/ns>.

VUTUKURU, M.; BALAKRISHNAN, H.; JAMIESON, K. Cross-layer wireless bit rate adaptation. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 39, n. 4, p. 3–14, 2009. ISSN 0146-4833.