

Uma Extensão ao Protocolo OLSR para Roteamento de Dados em Tempo Real Utilizando Múltiplos Caminhos

Andre J. G. Bessa
Laboratório de Redes de
Computadores e Segurança
(LARCES)
Universidade Estadual do
Ceará (UECE)
Fortaleza, CE, Brasil
andreb@larces.uece.br

Joaquim Celestino Junior
Laboratório de Redes de
Computadores e Segurança
(LARCES)
Universidade Estadual do
Ceará (UECE)
Fortaleza, CE, Brasil
celestino@larces.uece.br

Jorge L. Castro e Silva
Laboratório de Redes de
Computadores e Segurança
(LARCES)
Universidade Estadual do
Ceará (UECE)
Fortaleza, CE, Brasil
jlcs@larces.uece.br

Andre Ribeiro Cardoso
Laboratório de Redes de
Computadores e Segurança
(LARCES)
Universidade Estadual do
Ceará (UECE)
Fortaleza, CE, Brasil
andrec@larces.uece.br

ABSTRACT

It is increasingly common to use systems capable of transmitting multimedia content such as audio and video, in real-time in frainstructured networks, but it is not the same with Mobile Ad Hoc networks (MANET). A MANET is characterized by the free movement of its vehicles making packet routing with time delivery restrictions a complex problem. To solve this problem some works they propose the use of routing protocols with support for multiple paths. In this paper, we describe an extension to the Protocol OLSR, which has the goal to make it possible to support multimedia data transmission by multipath and use the data transmission quality present in RTCP reports to determine how many paths and which of them should be used to perform a real-time transmission in the most appropriate way.

RESUMO

É cada vez mais comum o uso de sistemas capazes de transmitir conteúdo multimídia, como áudio e vídeo, em tempo real em redes infraestruturadas, mas não podemos falar o mesmo das redes Ad Hoc móveis. Uma rede Ad Hoc móvel se caracteriza pela liberdade de movimentação dos seus nós tornando o roteamento de pacotes com restrições de tempo de entrega um problema complexo. Para resolver esse problema alguns trabalhos propõem o uso de protocolos de roteamento com suporte a múltiplos caminhos. Neste trabalho, descrevemos uma extensão ao protocolo OLSR, que possui

como objetivo torná-lo capaz de suportar a transmissão de dados multimídia por múltiplos caminhos e utilizar as informações de qualidade de transmissão presentes nos relatórios RTCP para determinar quantos caminhos e quais devem ser utilizados para realizar uma transmissão em tempo real de uma forma adequada. Essa extensão, chamada de OLSR MCTR, teve seu desempenho comparado, em diversos cenários, ao dos protocolos OLSR, AODV e AOMDV, através de simulações realizadas no simulador de redes ns2. Os resultados obtidos neste trabalho demonstram a viabilidade da nossa proposta.

Categories and Subject Descriptors

C.2.2 [Network Protocols]: Routing protocols

General Terms

Management, Performance

Keywords

Real Time, Ad Hoc, Multipath, OLSR, RTP, RTCP, MCTR

1. INTRODUÇÃO

As redes Ad Hoc móveis são redes móveis sem fio e sem uma infraestrutura. É um grande desafio fornecer serviços de comunicação multimídia, comunicações por áudio e vídeo em tais redes.

A transmissão de conteúdo multimídia em tempo real, como áudio e vídeo, exige que os protocolos envolvidos sejam capazes de atender a algumas restrições. Segundo a recomendação G.114 do *International Telecommunication Union* (ITU) [1], uma transmissão de áudio de boa qualidade exige um atraso máximo de 150 ms, sendo aceitável até 400 ms e a variação do atraso não pode ser superior a 1 ms.

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Full Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9642.
SBC - Brazilian Computer Society

O protocolo mais utilizado para transmitir esse tipo de tráfego é o *Real Time Transport Protocol* (RTP) em conjunto com o *Real Time Control Protocol* (RTCP) [2]. O RTP realiza o envio dos dados enquanto o RTCP monitora a qualidade da transmissão. Em redes infraestruturadas, o RTP apresenta um excelente funcionamento, porém em redes Ad Hoc o seu uso ainda está sendo desenvolvido.

As redes Ad Hoc móveis são redes formadas por dispositivos que não possuem ligações diretas entre si e são capazes de locomover-se. Nessas redes, cada um dos dispositivos funciona como um roteador com alcance limitado de transmissão. Devido ao constante deslocamento dos nós, a sua topologia sofre constantes alterações.

A constante modificação na topologia das redes Ad Hoc móveis torna complexo o roteamento do fluxo de dados. Portanto rotas que atendiam a um fluxo de dados são perdidas e novas rotas devem ser calculadas. Assim, quanto maior a dificuldade na determinação de um caminho, maior o atraso e pior o desempenho das aplicações [3]. Dessa forma, torna-se desafiador o uso dos protocolos RTP/RTCP, conforme concebidos, em redes Ad Hoc móveis, e novas propostas tem sido motivo de pesquisa na comunidade de redes [4].

Este trabalho propõe uma extensão ao protocolo *Optimized Link State Routing* (OLSR) [5] chamado aqui de OLSR Multicaminhos para Tempo Real (OLSR MCTR). Essa extensão torna o OLSR capaz de prover um mecanismo eficiente de roteamento para transmissões de dados em tempo real. O OLSR MCTR calcula múltiplas rotas até o destino, diferentemente do OLSR que calcula apenas uma. Por meio do uso de métricas de roteamento, os caminhos são ordenados e então parte deles é selecionada para realizar a transmissão de dados. Essa seleção é realizada por meio de uma técnica que utiliza as informações presentes nos relatórios RTCP para definir quantas rotas devem ser utilizadas durante uma transmissão. Como essa estratégia utiliza informações de um protocolo presente em uma camada diferente daquela onde se encontra o OLSR, ela é classificada como uma técnica *cross layer*.

A técnica *cross layer* consiste em permitir que os dados dos pacotes sejam tratados por mais de uma camada, e com isso aperfeiçoar o funcionamento dos protocolos. Essa técnica vem sendo utilizada em vários trabalhos, [6] [7], envolvendo redes ad hoc. Segundo o modelo OSI [8], 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.

Para avaliar a eficiência da nossa abordagem, o desempenho do OLSR MCTR foi comparado ao OLSR, ao *Ad hoc On-Demand Distance Vector* (AODV) e ao *Ad hoc On Demand Multipath Distance Vector* (AOMDV).

Este trabalho está dividido da seguinte forma: A seção II apresenta os trabalhos relacionados; A seção III apresenta uma breve descrição dos protocolos AODV, AOMDV, OLSR e RTP/RTCP; A seção IV apresenta o OLSR MCTR; A descrição da metodologia adotada para a realização das simulações seguida pela análise dos resultados obtidos é feita na seção V; e a seção VI conclui o presente artigo.

2. TRABALHOS RELACIONADOS

Em [9], foi definido 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 desempenho superior ao do AODV e ao AOMDV.

A estratégia para a seleção e uso de múltiplos caminhos adotada pelo OLSR MCTR baseia-se na análise dos dados presentes nos relatórios RTCP e na análise dos enlaces que ligam o nó a seus vizinhos.

Em [10], os autores propuseram o *Multiflow Real Time Transport Protocol* (MRTP), uma variação do RTP que particiona o fluxo de dados por múltiplos caminhos e com isso consegue reduzir a dependência de curto alcance na transmissão de dados multimídia, ou seja, transmissão de fluxo de dados em tempo real, que tradicionalmente sofrem degradação com o aumento no número de saltos, tem seu desempenho melhorado. Diferentemente da estratégia adotada pelo MRTP, o OLSR MCTR não particiona os pacotes entre os vários caminhos, ele particiona o fluxo de dados pelos diversos caminhos [11].

3. PROTOCOLOS UTILIZADOS NO TRABALHO

Para o desenvolvimento do OLSR MCTR foram estudados os protocolos OLSR e RTP/RTCP. A seguir o OLSR MCTR teve o seu desempenho comparado com o dos protocolos OLSR, AODV e AOMDV. Esta seção apresenta um resumo do funcionamento de cada um deles.

3.1 Optimized Link State Routing (OLSR)

O OLSR é um protocolo de roteamento proativo, periodicamente ele troca informações a respeito da rede para atualizar suas tabelas de rotas. Normalmente, um protocolo proativo ao receber uma mensagem com informações da rede, encaminha-a para outros nós ao seu alcance através de broadcast o que pode sobrecarregar a rede.

Para evitar isso, o OLSR usa os chamados *Multipoint Relay* (MPR) para encaminhar mensagens. *Multipoint Relaying* é uma técnica de redução do número de retransmissões redundantes de pacotes de controle em uma rede. Ela basicamente restringe o conjunto de nós retransmissores de todos para um subconjunto menor chamado de conjunto MPR [12].

O primeiro passo para montar um conjunto MPR é montar um conjunto contendo os vizinhos *one-hop* e outro com os vizinhos *two-hop*. O critério de seleção buscará o menor número de vizinhos *one-hop* capazes de alcançar todos os vizinhos *two-hop*, os nós selecionados formam então o conjunto MPR.

As duas mensagens de controle do OLSR são as mensagens *HELLO* e as mensagens de controle de topologia (TC). A primeira é utilizada para um nó trocar informações com seus vizinhos e verificar a qualidade da conexão. A segunda é

R_dest_addr	R_next_addr	R_dist	R_iface_addr
-------------	-------------	--------	--------------

Figura 1: Formato de uma entrada na tabela de roteamento do OLSR

usada para prover aos nós da rede informações a respeito da topologia da rede. As mensagens TC são enviadas em modo *broadcast* e os MPRs do nó remetente irão retransmitir essa mensagem, caso ele a tenha recebido pela primeira vez, caso contrário ela é descartada.

A tabela de rotas do OLSR é calculada a partir das informações dos enlaces locais e de topologia existente na base de dados do nó. Inicialmente, são incluídas as rotas 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. Quando ocorre alguma alteração dessas informações, a tabela tem de ser recalculada. A Figura 1 representa uma entrada na tabela de rotas do OLSR.

O campo `R_dest_addr` representa o destino localizado a distância `R_dist`, representada pelo número de saltos. O campo `R_next_addr` representa o próximo salto para se chegar em `R_dest_addr` e para acessá-lo deve-se enviar o fluxo de dados pela interface `R_next_addr`.

O algoritmo original de cálculo de rotas do OLSR busca as rotas com o menor número de saltos até o destino, porém essa abordagem não garante que essas rotas sejam boas, pois pode existir algum enlace com baixa capacidade de transmissão ao longo das rotas. Para prevenir esse problema, algumas implementações utilizam métrica de roteamento como a *Expected Transmission Count (ETX)* [13].

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, e com isso encontrar as rotas com a melhor vazão. No OLSR, o ETX de um enlace é calculado em função da quantidade de mensagens *HELLO* trocadas com os vizinhos em comparação com a quantidade esperada. Os nós que compartilham um enlace trocam entre si o valor desse cálculo. O valor obtido por um nó é chamado de *Link Quality (LQ)* e o valor obtido pelo vizinho é chamado de *Neighbor Link Quality (NLQ)*. Esses dois valores então são aplicados na fórmula 1. Quanto menor o valor do ETX, melhor para a rota.

$$ETX = \frac{1}{LQL * NLQ} \quad (1)$$

3.2 Real-Time Transport Protocol/ Real-Time Transport Control Protocol (RTP/RTCP)

O RTP e o RTCP foram desenvolvidos pelo *Audio and Video Transport Working Group* do IETF [14] e definidos na RFC 3550 [15]. 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 [16]. 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*.

Cada participante de uma sessão RTP envia pacotes contendo aproximadamente 20 ms de conteúdo encapsulado em um cabeçalho RTP. Esses cabeçalhos são encapsulados por um protocolo da camada de transporte, normalmente o UDP. Ao chegarem ao destino, os pacotes são armazenados e organizados no *playout buffer* (ou *jitter buffer*). O *playout buffer* tem o objetivo de garantir a reprodução síncrona e contínua da conversa entre os participantes de uma sessão RTP. Assim ele absorve as variações do atraso dos pacotes e descarta aqueles fora do seu *playout*, ou seja, fora do intervalo de tempo que deveria ter sido reproduzido. Assim, percebe-se uma relação direta entre o atraso e a perda de pacotes RTP [17].

O RTP não possui nenhum mecanismo para averiguar se a transmissão está ocorrendo da forma desejada. Para esse fim, foi criado o RTCP (*Real-Time Transport Control Protocol*). O RTCP baseia-se em transmissões periódicas de pacotes de controle, chamados de relatórios RTCP, entre os participantes de uma sessão RTP. O RTCP possui dois tipos de relatórios, o RTCP *Sender Report (RTCP SR)* e o RTCP *Receiver Report (RTCP RR)*. O RTCP SR é o relatório enviado do remetente para o destino e o RTCP RR é o relatório enviado do destinatário para o remetente.

O RTCP foi criado para monitorar as sessões RTP, provendo relatórios sobre a qualidade da transmissão. Periodicamente os participantes de uma sessão RTP trocam mensagens contendo informações como o número de pacotes perdidos, o valor do atraso da transmissão, e a sua variação. Os pacotes RTCP são transmitidos pelo mesmo canal do RTP e a sua taxa de transmissão é controlada para não ocasionar perda na qualidade de transmissão de dados RTP.

As informações presentes nos relatórios RTCP são usadas para realizar ajustes nas aplicações transmissoras, tais como alterar o encoder de áudio e vídeo. Entretanto, se o motivo da baixa qualidade de transmissão encontra-se na camada de rede, essas informações podem não ajudar a minimizar esse problema. Nas redes infraestruturas, essa estratégia pode passar despercebida, mas o dinamismo presente nas redes Ad Hoc torna-a relevante no processo de roteamento de dados.

3.3 Ad hoc On Demand Distance Vector (AODV)

O AODV é um protocolo de roteamento reativo baseado no algoritmo de vetor de distância [18]. O AODV elimina a necessidade de um *broadcast* global das informações de roteamento, evitando o alto consumo de banda por pacotes de controle.

Para descobrir uma rota até um destino, são enviadas as mensagens *Route REQuest (RREQ)* em *broadcast*. Cada RREQ possui um identificador, que junto com o endereço IP do nó origem, diferencia as mensagens RREQs. Um nó ao receber uma mensagem RREQ, caso seja o destino, ou

conheça uma rota para ele, envia a mensagem *Route REPLY* (RREP) para o nó de origem, caso contrário, o nó faz a solicitação para seus nós vizinhos. Cada nó conhece apenas o próximo salto e a quantidade de saltos para se alcançar o destino.

Para a identificação rápida de quebra de enlace, mensagens *HELLO* são trocadas periodicamente entre vizinhos. Quando um nó deixa de receber essas mensagens de um vizinho, ele assume que o enlace entre ele e o vizinho não está mais ativo, então uma mensagem chamada de *Route Error* (RERR) é enviada para os nós que dependiam desse enlace para estabelecer uma rota.

3.4 Ad hoc On Demand Multipath Distance Vector (AOMDV)

O AOMDV [19] é uma extensão do AODV, cuja 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.

Tal qual o AODV, o AOMDV transmite mensagens RREQ pela rede em busca de novas rotas até certo destino e recebe como resposta uma mensagem RREP. Foi adicionada a mensagem RREQ original um campo a mais chamado de *firsthop* que armazena um identificador do nó vizinho por onde a mensagem passou inicialmente. Essa informação será armazenada em uma lista presente em cada nó chamada de *firsthop_list*.

Quando o AOMDV recebe um RREQ duplicado ele não o descarta imediatamente como o AODV, primeiro ele verificará se o campo *firsthop* do nó já se encontra na sua *firsthop_list*, caso não, a mensagem RREQ poderá ser respondida, ou encaminhada para o seus vizinhos. Dessa forma pelo menos um nó no caminho será diferente, garantindo assim a disjunção por enlace.

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. Quando o AODV detecta uma rota interrompida durante uma transmissão de dados, ele executa novamente o mecanismo de descoberta de rotas. Utilizando múltiplos caminhos, caso alguma rota caia, basta concentrar a transmissão de dados em outra rota disponível, assim o mecanismo de descoberta de rotas só será chamado em caso de todas as rotas disponíveis se tornarem inválidas [19].

4. O OLSR MCTR

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 nele implementada e modificando algumas das suas estruturas de dados.

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.

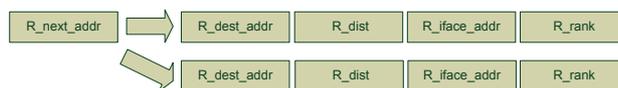


Figura 2: Formato de uma entrada na tabela de roteamento do OLSR MCTR



Figura 3: Formato de uma entrada na TGR

Já o OLSR MCTR implementa uma tabela de rotas capaz de armazenar mais de uma rota para cada destino, a Figura 2 apresenta o formato de uma entrada na tabela de rotas do OLSR MCTR. Cada entrada na tabela representa um destino e cada destino possui uma lista de caminhos possíveis.

Definimos que cada entrada na lista deve possuir quatro campos. Os três primeiros foram baseados nos campos originais definidos no OLSR e o último, *R_rank*, foi criado para o OLSR MCTR. O *R_rank* é utilizado para inferir a prioridade de cada rota. O valor de prioridade é calculado em função da métrica de roteamento ETX, dessa forma é possível ordenar as rotas em função da qualidade de seus enlaces.

Além de uma nova tabela de roteamento, o OLSR MCTR implementa uma nova estrutura auxiliar chamada de Tabela de Gerenciamento de Rotas (TGR). Essa tabela contém informações responsáveis por definir por quantos caminhos os pacotes RTP serão encaminhados e quais serão estes caminhos. Uma entrada na tabela de gerenciamento de rotas tem o formato descrito na Figura 3.

O campo *RTABLE_SIZE* representa a quantidade de rotas existentes para um dado caminho. O campo *PATH_COUNT* define qual rota deverá ser utilizada para a transmissão de um determinado pacote. O campo *OLSR_MCTR_COUNT* representa o número de rotas que devem ser utilizadas no processo de transmissão de dados. O seu valor é calculado pelo algoritmo NIR, conforme definido na próxima seção.

4.1 Número Ideal de Rotas (NIR)

O algoritmo NIR, descrito pelo algoritmo 1, recebe como dados de entrada o valor do atraso fim-a-fim da transmissão corrente. Esse valor representa o tempo que um pacote demora para sair do nó origem e chegar ao nó destino.

O valor do atraso fim-a-fim é obtido a partir do *Round-Trip Time* (RTT) de um pacote na rede e por sua vez o RTT é obtido a partir dos dados fornecidos pelos relatórios RTCP RR.

O OLSR MCTR ao receber um pacote RTCP RR busca em seu cabeçalho os campos LSR (*Last Sender Report*) e DLSR, (*Delay Last Sender Report*). Esses campos representam respectivamente o momento em que o nó de origem do pacote RTCP RR recebeu pela última vez um pacote RTCP SR do nó corrente e o intervalo de tempo entre 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.

Para obter esse valor subtrai-se do tempo corrente o valor presente no campo LSR e do resultado subtrai-se o valor do campo DLSR, o resultado será o RTT. Para obter o valor do atraso fim-a-fim basta dividir o valor do RTT por 2 e então será obtido o tempo do atraso em um dos trechos.

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 está ocorrendo segundo as restrições propostas pela recomendação G 114.

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 encontra-se acima de 300 ms, o algoritmo NIR atualiza o valor de OLSR_MCTR_COUNT.

Algoritmo 1 Número Ideal de Rotas

```

1: if atraso_fim_fim ≥ 300ms then
2:   if ultima_verificacao = 0 then
3:     ultima_verificacao ← tempo_corrente
4:     parar
5:   end if
6:   if tempo_corrente – ultima_verificacao > 500ms
   then
7:     OLSR_MCTR_COUNT ←
       OLSR_MCTR_COUNT + 1
8:     ultima_verificacao ← tempo_corrente
9:   end if
10: else if 300ms > atraso_fim_fim > 100ms then
11:   ultima_verificacao ← tempo_corrente
12: else if 100ms < atraso_fim_fim &
   OLSR_MCTR_COUNT > 1 then
13:   if tempo_corrente – ultima_verificacao > 10 then
14:     OLSR_MCTR_COUNT ←
       OLSR_MCTR_COUNT – 1
15:     ultima_verificacao ← tempo_corrente
16:   end if
17: end if

```

4.2 Encaminhamento de Pacotes

O Algoritmo 2 representa o funcionamento do mecanismo de transmissão de dados do OLSR MCTR. Quando o OLSR MCTR deseja transmitir um fluxo de dados RTP ele busca na TGR o campo RTABLE_SIZE para o destino almejado, caso o seu valor seja superior a 1 significa que a tabela de roteamento possui mais de um caminho até o destino.

Algoritmo 2 Encaminhamento de Pacotes

```

1: RTABLE_SIZE ← numero_rotas_destino
2: if PATH_COUNT > OLSR_MCTR_COUNT
   OR PATH_COUNT > RTABLE_SIZE OR
   PATH_COUNT = 0 then
3:   PATH_COUNT ← 1
4: end if
5: emcaminhar_pacote_pela_rota(PATH_COUNT)
6: PATH_COUNT ← PATH_COUNT + 1

```

O próximo campo observado é o OLSR_MCTR_COUNT. O seu valor inicial é 1. Caso o algoritmo NIR considere que a

qualidade da transmissão não é adequada, o seu valor é incrementado. O campo PATH_COUNT define qual rota será utilizada para o envio de cada pacote. A rota é definida em função da sua prioridade. Caso o seu valor seja 1, a rota com a maior prioridade, segundo o campo R_rank, será utilizada. Em seguida seu valor é incrementado para 2, caso ele continue inferior ou igual a RTABLE_SIZE e inferior ou igual a OLSR_MCTR_COUNT, a rota utilizada para encaminhar o próximo pacote será aquela com a segunda maior prioridade, caso o contrário, o valor de PATH_COUNT voltará a ser 1. Isso faz com que os caminhos sejam utilizados de forma alternada.

5. AVALIAÇÃO DE DESEMPENHO

O OLSR MCTR teve sua desempenho comparada aos protocolos OLSR, AODV e AOMDV por meio de simulações implementadas no simulador de redes *Network Simulator 2* (NS-2) [20]. 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 [21]. A implementação do AODV já acompanha o pacote do NS-2. A implementação do AOMDV utilizada foi desenvolvida por [22]. A implementação do RTP/RTCP pertence ao pacote padrão do NS-2 e foi atualizada por [23].

Os cenários e o tráfego de fundo foram gerados a partir do programa BonnMotion [24] O modelo de mobilidade dos nós adotado foi o *Random Waypoint*. Cada cenário foi executado 33 vezes. A configuração dos cenários é descrita na Tabela 1.

As métricas escolhidas para avaliar os protocolos foram o atraso fim-a-fim, a variação do atraso e a relação entre o número de *bytes* recebidos e *bytes* enviados.

Tabela 1: Configurações utilizadas para a realização das simulações

Parâmetros das Simulações	
Área do Cenário	1000 m x 1000 m
Tempo de Simulação	200 s
Aplicação	<i>Constant Bitrate</i> (CBR)
MAC	802.11Ext
PHY	WirelessPhyExt
Modelo de Propagação	TwoRayGround
Número de execuções	33
Número de nós	50
Velocidade dos Nós	5 m/s, 10 m/s, 12 m/s, 15 m/s, 20 m/s
Alcance do sinal	250 m

5.1 Atraso Fim-a-Fim

A Figura 4 apresenta o gráfico com os resultados do atraso fim-a-fim para cada um dos protocolos nos cenários propostos. O OLSR MCTR e o AOMDV apresentaram um melhor comportamento em comparação com os outros protocolos.

Isso ocorre pois o aumento da velocidade de deslocamento dos nós diminui o tempo de existência de uma rota devido à alteração mais rápida dos enlaces de cada nó. Os protocolos projetados para calcular uma rota única para cada

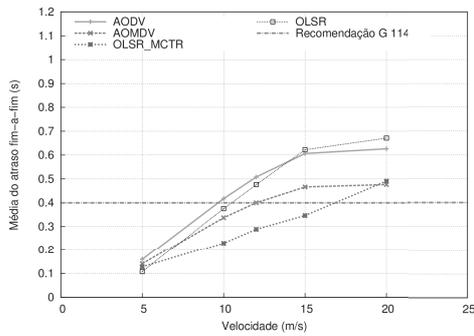


Figura 4: Valor médio do atraso fim-a-fim

destino necessitam atualizar a tabela de rotas sempre que uma alteração em uma rota é verificada. Já os protocolos projetados para calcular mais de uma rota para o destino são mais robustos a essa situação.

O OLSR MCTR e o AOMDV não sofreram tanto quanto os outros dois protocolos, pois não necessitam atualizar suas tabelas de rotas com a mesma frequência do OLSR e do AODV. A Figura 4 demonstra que o atraso fim-a-fim do OLSR MCTR, em comparação com os outros protocolos, aumenta de forma mais lenta com a variação da velocidade. Dessa forma, ele permaneceu durante a maior parte do experimento abaixo dos 400 ms, limite estabelecido pela recomendação G 114 para a transmissão de dados em tempo real.

5.2 Variação do Atraso

A Figura 5 representa o comportamento da variação do atraso de cada protocolo ao longo de cada simulação. Segundo o gráfico, à medida que a velocidade de deslocamento dos nós aumenta também aumenta a variação do atraso.

A Figura 6 representa o uso médio de caminhos pelo AOMDV e pelo OLSR MCTR. A medida que a velocidade dos nós aumenta, aumenta o uso de múltiplos caminhos e esse uso acompanha o aumento da variação do atraso. Isso ocorre pois transmissões paralelas de dados interferem na ordem com que os pacotes chegam até o destino. Como cada caminho é percorrido com uma velocidade diferente por cada pacote, 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, como é possível observar na Figura 6.

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

5.3 Taxa de Pacotes Entregues

A Figura 7 representa a porcentagem de dados efetivamente entregues em cada um dos cenários. A diminuição da taxa de entrega acompanha o aumento no atraso fim-a-fim. Devido

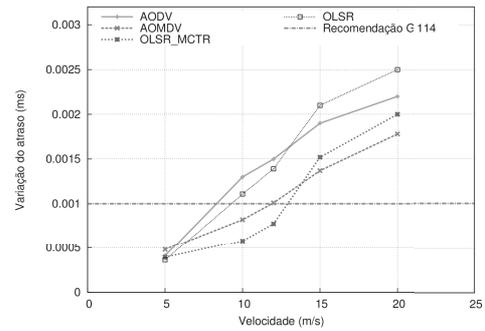


Figura 5: Valor médio de variação do atraso

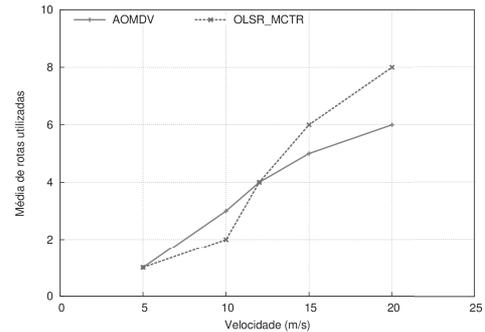


Figura 6: Valor médio de rotas utilizadas

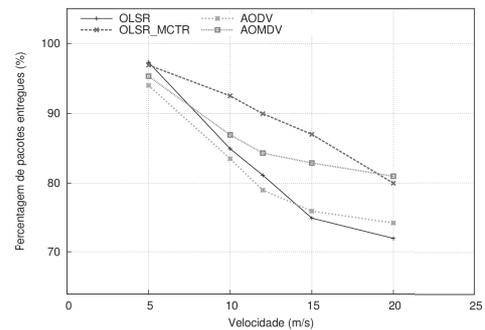


Figura 7: Valor médio de pacotes entregues

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.

6. CONCLUSÃO E TRABALHOS FUTUROS

Esse trabalho apresentou uma extensão ao protocolo OLSR chamada de OLSR MCTR. Essa extensão torna o OLSR capaz de utilizar múltiplas rotas para o encaminhamento de dados em tempo real. Para gerenciar o uso das rotas, o OLSR MCTR utiliza informações disponíveis nos relatórios RTCP. Para avaliar a viabilidade dessa proposta o OLSR MCTR teve seu desempenho comparado com os protocolos OLSR, AODV e AOMDV. Os resultados obtidos demonstraram um comportamento superior do OLSR MCTR em comparação aos demais protocolos. Os dados também deixaram claro que o uso de uma estratégia de múltiplos ca-

minhos para roteamento melhora o desempenho do RTP em redes Ad Hoc móveis, mas essa estratégia deve ser utilizada com cautela, pois ele pode ocasionar aumento da variação do atraso, o que não é interessante para a transmissão de dados em tempo real.

O OLSR MCTR possui alguns pontos a serem melhorados. Seria interessante avaliar o uso de outras métricas de roteamento no processo de classificação de rotas no lugar da ETX tais como algumas de suas extensões, como a *Expected Transmission Time* (ETT) [25] e como a *Modified Expected Number of Transmissions* (mETX) [26]. O algoritmo NIR executa um procedimento que reduz o número de caminhos a serem utilizados após um dado tempo. Essa técnica pode ser melhorada com o uso de informações sobre a topologia da rede contidas nas mensagens TC do OLSR.

7. REFERÊNCIAS

- [1] I. Recommendation. G. 114. *One way Transmission Time*, 1996.
- [2] A. Abd El Al, T. Saadawi, and M. Lee. Unequal error protection for real-time video in mobile ad hoc networks via multi-path transport. *Comput. Commun.*, 30(17):3293–3306, 2007.
- [3] S. Mao, S. Lin, Y. Wang, S. Panwar, and Y. Li. Multipath video transport over ad hoc networks. *IEEE Wireless Communications*, 12(4):42–49, 2005.
- [4] C. Perkins. Ad hoc networking: An introduction. *Ad Hoc Networking*, pages 314–318, 2001.
- [5] T. Clausen and P. Jacquet. RFC3626: Optimized Link State Routing Protocol (OLSR), 2003.
- [6] T. Li, H. Ji, J. Mei, Y. Li, and C. Hu. 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*, pages 2943–2947, Piscataway, NJ, USA, 2009. IEEE Press.
- [7] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. *SIGCOMM Comput. Commun. Rev.*, 39(4):3–14, 2009.
- [8] J. Day. The (un)revised osi reference model. *SIGCOMM Comput. Commun. Rev.*, 25(5):39–55, 1995.
- [9] T. Nguyen, S. W. Loke, T. Torabi, and H. Lu. Placesense: a tool for sensing communities. In *Proceedings of the 4th international conference on Wireless pervasive computing, ISWPC'09*, pages 263–267, Piscataway, NJ, USA, 2009. IEEE Press.
- [10] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar. Mrtp: a multiframe real-time transport protocol for ad hoc networks. *IEEE Transactions on Multimedia*, 8(2):356–369, 2006.
- [11] Q. Lu, L. Du, Z. Zuo, and X. Xiao. Improved multi-path aodv protocols for real-time video transport over mobile ad hoc networks. *Pacific-Asia Workshop on Computational Intelligence and Industrial Application, IEEE*, 1:621–625, 2008.
- [12] A. Laouiti, A. Qayyum, and L. Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks, 2001.
- [13] X. Ni, K.-c. Lan, and R. Malaney. On the performance of expected transmission count (etx) for wireless mesh networks. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools, ValueTools '08*, pages 77:1–77:10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [14] IETF. Ietf audio and video transport working group, July 2011.
- [15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rfc3550 – rtp: A transport protocol for real-time applications, 2003.
- [16] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach*. Addison-Wesley Publishing Company, USA, 5th edition, 2009.
- [17] C. A. C. MARCONDES, R. P. H. de Aguiar, D. Fábio, and P. J. C. Ambiente para simulação e monitoração de ligações telefônicas ip. *XXI Simpósio Brasileiro de Redes de Computadores*, 2003.
- [18] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.
- [19] N. T. Nguyen, A.-I. A. Wang, P. Reiher, and G. Kuenning. Electric-field-based routing: a reliable framework for routing in manets. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):35–49, 2004.
- [20] The network simulator - ns-2.
- [21] F. J. Ros. Um-olsr. MANET Simulation and Implementation at the University of Murcia, 2009.
- [22] S. Das and M. Marina. Aomdv. CMU/MONARCH group Carnegie Mellon University, 2004.
- [23] C. Bouras, A. Gkamas, and G. Kioumourtzis. Extending the functionality of rtp/rtcp implementation in network simulator (ns-2) to support tcp friendly congestion control. In *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–6, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [24] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. BonnMotion: a mobility scenario generation and analysis tool. In *SIMUtools '10: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, pages 1–10, ICST, Brussels, Belgium, Belgium, 2010. ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering).
- [25] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking, MobiCom '04*, pages 114–128, New York, NY, USA, 2004. ACM.
- [26] C. E. Koksal, K. Jamieson, E. Telatar, and P. Thiran. Impacts of channel variability on link-level throughput in wireless networks. *SIGMETRICS Perform. Eval. Rev.*, 34:51–62, June 2006.