

OLSR Fuzzy Cost (OLSR-FC): an extension to OLSR protocol based on fuzzy logic and applied to avoid selfish nodes

OLSR Custo Fuzzy: uma extensão ao protocolo OLSR baseada em lógica Fuzzy e aplicada à prevenção de nós egoístas em MANETs

Diógenes Antonio Marques José^{1*}, Renato F. Bulcão-Neto², Vinícius Sebba Patto² e Iwens Gervásio Sene Júnior²

Abstract: The mobile ad-hoc networks (MANET) are those whose nodes have mobility, energy restriction and operate simultaneously as end systems and router. One of the main problems found in MANETs is the occurrence of selfish nodes, which are those that refuse to route packets for other nodes. To address the issue of selfish nodes in MANETs and improve the flow of traffic in these networks, this paper proposes an extension to the OLSR protocol, based on Fuzzy logic, called OLSR Fuzzy Cost (OLSR -FC). Using the NS-2 simulator, the OLSR-FC proposal was compared to other extensions of OLSR protocol (e.g., OLSR-ETX, OLSR-ML e OLSR-MD) concerning the performance metrics: packet loss, end-to-end delay, Jitter, power consumption, routing overhead and throughput. The results showed that OLSR-FC obtains better performance than the evaluated extensions, avoiding selfish nodes and selecting routes whose links have little packet losses.

Keywords: MANETS — Selfish Nodes — Routing — OLSR — Fuzzy

Resumo: As redes móveis ad-hoc (MANETs) são aquelas cujos nós possuem mobilidade, restrição de energia e que funcionam simultaneamente como sistemas finais e roteador. Um dos principais problemas encontrados em MANETs é a ocorrência de nós egoístas, aqueles que se recusam a rotear pacotes em favor de outros nós. Para tratar o problema dos nós egoístas em MANETs e melhorar o fluxo do tráfego nessas redes, este trabalho propõe uma extensão ao protocolo OLSR, baseada em lógica Fuzzy, denominada OLSR Fuzzy Cost (OLSR-FC). Utilizando o simulador NS-2, a proposta foi comparada a outras extensões do protocolo OLSR (ex., OLSR-ETX, OLSR-ML e OLSR-MD) quanto às métricas de desempenho: perda de pacotes, atraso fim-a-fim, Jitter, consumo de energia, overhead de roteamento e vazão. Os resultados mostraram que o OLSR-FC obtém melhor desempenho que as extensões avaliadas, evitando nós egoístas e escolhendo rotas cujos enlaces possuem poucas perdas de pacotes.

Palavras-Chave: MANETS — Selfish Nodes — Routing — OLSR — Fuzzy

¹Universidade do Estado de Mato Grosso (UNEMAT), Brasil

²Universidade Federal de Goiás (UFG), Brasil

*Corresponding author: dioxfile@unemat.br

DOI: <https://doi.org/10.22456/2175-2745.86380> • Received: 02/09/2018 • Accepted: 20/02/2019

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

1. Introdução

A difusão em massa do padrão de comunicação IEEE 802.11 fez das redes móveis *ad-hoc* (MANETs)¹ um dos temas de pesquisa mais promissores [1]. Dentre os assuntos relacionados às MANETs destacam-se os protocolos de roteamento [2], qualidade de serviço [3] e segurança [4]. As principais características dessas redes são: ausência de infraestrutura, capacidade de realizar roteamento multi-salto, mobilidade,

auto-gerenciamento, auto-organização e restrição de energia [5]. Uma das principais vantagens das MANETs consiste no fato de que todos os nós atuam, ao mesmo tempo, como *sistemas finais* e roteador. Há diversas situações nas quais as MANETs podem ser aplicadas como: operações emergenciais de busca e resgate em ambientes inóspitos; comunicação entre automóveis em trânsito; e uso local em conferência ou sala de aula (ex., troca rápida de informação por meio de *notebook*, *smartphones*, etc). Um dos principais problemas nessas redes que está diretamente relacionado à restrição de energia,

¹Do inglês, *Mobile Ad-Hoc Networks (MANETs)*.

consiste no comportamento egoísta dos nós, que pode ser definido pela atitude de obter vantagens da rede (ex., vazão, *watts*, etc) [6], principalmente no encaminhamento de pacotes, que consome uma quantidade significante de energia e nem todos os nós estão dispostos a gastar esse precioso recurso encaminhando pacotes que não sejam seus [7]. A ação do nó egoísta nas *MANETs* pode causar diversos problemas, dentre eles destacam-se: altas taxas de perdas de pacotes; aumento do número de *enlaces* falhos e redução dos pacotes redirecionados. Na prática desabilitar o encaminhamento de pacotes em nome de outros nós pode ser facilmente feito em sistemas *Linux*, basta apenas digitar o seguinte comando no terminal: `echo 0 > /proc/sys/net/ipv4/ip_forward`.

Para resolver os problemas citados acima, diversos mecanismos já foram propostos, porém a maioria deles forçam a cooperação, isto é, ou o nó coopera redirecionando pacotes em favor de outros nós ou ele será punido e ficará com o acesso aos serviços de redes limitado [8]. Segundo Toh *et al.* [9], a desvantagem da punição consiste no fato de que para punir completamente um nó egoísta é necessário coordenar punições colaborativas no meio sem fio, o que ocasionaria um alto tráfego de mensagens de controle (*overhead*). Além disso, muitos desses mecanismos podem cometer falhas, falsos positivos para o egoísmo. Por exemplo, o mecanismo *ICARUS* não consegue efetivamente detectar nós egoístas que não atuam dessa forma o tempo todo [10]. Portanto, os sistemas que usam a punição como método principal de combate à ação egoísta em *MANETs* seguem o caminho inverso do paradigma das mesmas, o encaminhamento cooperativo de pacotes.

Este artigo apresenta uma métrica de roteamento baseada em lógica *Fuzzy*² com múltiplos parâmetros (ex., *tакса de perda de pacotes, energia e conectividade*) implementada no protocolo *OLSR* [12] e denominada *OLSR Fuzzy Cost (OLSR-FC)*. Os principais objetivos do *OLSR-FC* consistem em minimizar o impacto dos nós egoístas e melhorar o desempenho do tráfego nas *MANETs*. Para a avaliação da métrica foi utilizado o simulador de redes *NS-2* [13] e foram realizadas exaustivas simulações em que o *OLSR-FC* foi comparado às extensões *OLSR Expected Transmission Count (OLSR-ETX)* [14], *OLSR Minimum Loss (OLSR-ML)* [15] e *OLSR Minimum Delay (OLSR-MD)* [16]. Na avaliação foram consideradas as seguintes métricas de desempenho: perda de pacotes, atraso fim-a-fim, *Jitter*, consumo de energia, *overhead* de roteamento e vazão. Os resultados mostraram que o *OLSR-FC* escolhe rotas cujos *enlaces* possuem poucas perdas de pacotes causadas por nós egoístas. Nesse sentido, seu desempenho foi superior às extensões avaliadas, por exemplo, o ganho do *OLSR-FC*, com relação à perda de pacotes no cenário com até 40% de nós

²O conceito de lógica *Fuzzy* foi criado por Lotfali Askar-Zadeh em seu trabalho denominado *Fuzzy Sets* em 1965 [11]. Este conceito é aplicado às classes de objetos do mundo real que não possuem um critério de definição de membros preciso. Por exemplo, conjuntos das pessoas obesas, que pode ter diferentes graus de obesidade. Nesse sentido, o ambiente *MANET* pode ser considerado impreciso, pois em muitas situações não há certeza quanto a alguns aspectos. Por exemplo, a conectividade nas *MANETs* pode ter diferentes níveis entre alto e baixo dependendo da posição em que o nó se encontra na rede.

egoístas foi de 97,89% em comparação à extensão do *OLSR* com o melhor desempenho (*OLSR-ETX*).

O restante deste artigo está dividido da seguinte forma: a Seção 2 descreve os trabalhos relacionados; a Seção 3 apresenta em detalhes a descrição da proposta; na Seção 4 são apresentados os materiais e métodos; a Seção 5 apresenta os resultados e a discussão; e finalmente, na Seção 6, são apresentadas a conclusão e as possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

2.1 Nós Egoístas em *MANETs*

Zayani e Zeghlache [5] descrevem que para manter a conectividade da rede, cada nó deve encaminhar pacotes para outros nós. Entretanto, se a energia residual do nó for limitada, ele tende a ser não cooperativo descartando os pacotes que deveria encaminhar. Este tipo de comportamento é descrito na literatura como comportamento egoísta.

Para Babakhouya *et al.* [17], o problema do nó egoísta consiste no fato de ele participar das funções de roteamento, concordando em encaminhar pacotes em nome de outros nós, mas, em vez disso, descarta os pacotes silenciosamente na tentativa de economizar a energia e a largura de banda. Nesta pesquisa são considerados dois tipos de nós egoístas: o que participa corretamente das funções de roteamento, porém não redireciona pacotes de dados (tipo 1); e o que não participa, descartando mensagens de controle, por exemplo, *RREQ*³ (tipo 2). Conforme abordado por Babakhouya *et al.*, o nó egoísta do tipo 2 não tem grande efeito na taxa de entrega de pacotes. Todavia, o tipo 1 é o mais prejudicial à rede por degradar a taxa de entrega de pacotes. Para sanar o problema é necessário o uso de recursos como: criptografia e monitoramento. Entretanto, estes mecanismos podem reduzir o desempenho do tráfego, adicionando sobrecarga à rede.

Conforme abordado por Yokoyama *et al.* [18], o comportamento egoísta é classificado em três categorias: *Deny of Service (DoS)*, negligente e ganancioso. Além disso, os autores afirmam que o tipo de nó egoísta mais prejudicial à rede é aquele que participa do processo de construção das rotas, porém, ao receber pacotes para serem redirecionados ele os descarta. Neste trabalho são propostos métodos para detectar os padrões de comportamento egoísta, entretanto, apesar de detectarem este padrão de comportamento, alguns métodos cometem falsos positivos. Por exemplo, a contramedida aplicada ao comportamento egoísta que descarta pacote de dados e retransmite mensagens de controle não teve nenhum efeito.

Toh *et al.* [9] propõem um *Mecanismo de Incentivo a Cooperação (MIC)*, baseado em negociação, chamado *Selfish Check Negotiation Protocol (SCNP)*. O *SCNP* usa o método explícito que pergunta aos nós da rede se eles concordam em encaminhar pacotes de outros. Além disso, é importante classificar o egoísmo de forma correta porque falsas detecções,

³*Route Request*, mensagem de solicitação de rota enviada por protocolos de roteamento reativos tais como *AODV* e *DSR*.

por exemplo, falsos positivos, podem gerar redes menos úteis, com rotas reduzidas. Da mesma forma, um falso negativo pode resultar no aumento da quantidade de pacotes descartados. A pesquisa ainda recomenda que a cooperação não deve ser forçada e sim negociada. O grande problema desta proposta consiste no fato de que ela não garante a cooperação porque o possível encaminhador de pacotes pode não concordar com os termos da negociação, ou simplesmente ignorar, e continuar agindo como nó egoísta.

Proposto por Robert *et al.* [8], o *Reputation-based Clustering (RBC-OLSR)* objetiva adicionar confiança e prolongar o tempo de vida das *MANETs* na presença de nós egoístas. A proposta gera duas novas extensões a *H-OLSR*, que é capaz de melhorar o tempo de vida da rede e a *RH-OLSR*, um modelo híbrido que provê um *MIC* baseado em reputação e recompensa. Um dos objetivos da proposta consiste em motivar os nós egoístas a atuarem como encaminhadores de pacotes⁴. A principal desvantagem da proposta, consiste no fato de ela oferecer acesso aos serviços de rede com base na reputação dos nós. Dessa forma, um nó não egoísta com baixa reputação pode ter o acesso à rede limitado.

2.2 Protocolo OLSR e Extensões

Esta seção apresenta uma visão geral do protocolo *OLSR* e de algumas extensões propostas a ele. Neste artigo, métrica de roteamento é definida como o critério utilizado pelo protocolo para selecionar rotas e extensão é definida como uma variação do protocolo que utiliza critério de seleção de rotas diferente da versão original.

O protocolo *OLSR*, descrito na *RFC 3626* [12], consiste em um protocolo de roteamento proativo, desenvolvido para uso em *MANETs* e *Wireless Mesh Networks (WMN)*. A sua principal vantagem consiste no uso de nós especializados denominados *Multipoint Relay (MPRs)*⁵ que têm a função de diminuir a quantidade de mensagens de controle duplicadas, essa característica torna o *OLSR* adequado para ser usado em redes com altas densidades de nós. Além disso, o *OLSR* usa como critério de escolha de rotas o número de saltos (*hop-count*), o qual é contabilizado por meio de um algoritmo de busca em largura. Este critério possui um custo computacional $O(|E| + |V|)$ que o torna eficiente computacionalmente, se comparado aos protocolos que usam o algoritmo de *Dijkstra* $O(|V|^2)$.

O *OLSR*, por ser um protocolo que enfatiza o tráfego do melhor esforço, pode escolher caminhos com baixa qualidade (ex., baixa vazão, atraso elevado, alta taxa de perda de pacotes, etc), e em função disso diversas extensões têm sido propostas para resolver esta desvantagem. Por exemplo, Ge *et al.* propõem o *QOLSR* [3], o qual possui três heurísticas de roteamento que permitem ao *OLSR* encontrar caminhos com a máxima largura de banda. Proposta por

⁴Cluster Head ou MPR.

⁵Os *MPRs* são selecionados por seus vizinhos, a um salto de distância, para difundir a topologia da rede por meio das mensagens *TC*. As mensagens *HELLO* são usadas para avisar os vizinhos a um salto sobre o estado dos enlaces e não são redirecionadas pelos *MPRs*.

D. Couto *et al.* [14], a métrica *ETX*⁶ tem como principal objetivo escolher rotas com o menor número esperado de transmissões/retransmissões (ex., $ETX = \frac{1}{d_f * d_r}$) necessárias para entregar *frames* com sucesso ao destino. A extensão *OLSR-ML*, desenvolvida por Passos *et al.* [15], tem como principal objetivo encontrar caminhos com a menor probabilidade de perda de pacotes (ex., $ML = d_f * d_r$). Desenvolvida por Cordeiro *et al.* [16], a extensão *OLSR-MD* tem como meta principal escolher rotas com o menor atraso de transmissão por meio do uso do *AdHoc Probe*, que utiliza pares de pacotes para medir o atraso do enlace em um único sentido (ex., $OWD = (T_{recv1,i} - T_{send,i} - \delta) - (T_{recv2,i} - T_{send,i} - \delta)$).

Gomes *et al.* [19], propõem uma extensão ao protocolo *OLSR*, baseada em lógica *Fuzzy* e denominada *OLSR-FLC (Fuzzy Link Cost)*. A extensão proposta tem o objetivo de melhorar o desempenho da distribuição de conteúdo multimídia em *WMN* levando em consideração dois critérios: *QoS* e *QoE*. O *Sistema de Inferência Fuzzy* proposto ao *OLSR-FLC* recebe como entrada duas métricas de roteamento conhecidas, *ETX* e *MD*. Sua avaliação foi feita no *NS-2* e os resultados mostraram um bom desempenho do *OLSR-FLC* com relação às métricas de desempenho atraso, *Jitter*, vazão e probabilidade de bloqueio (ex., perda de pacotes), no que se refere à *QoS*. Além disso, esta proposta também apresentou um desempenho superior, no cenário proposto, ao das outras extensões avaliadas (ex., *OLSR*, *OLSR-ETX*, *OLSR-MD* e *OLSR-DC*) nos quesitos *Peak Signal to Noise Ratio (PSNR)*, usado para comparar *frame* a *frame* a qualidade do vídeo recebido pelo usuário com o vídeo original, e *Mean Opinion Score (MOS)*, que foi usado para estimar a qualidade de aplicações multimídia, baseada na opinião do usuário (ex., *QoE*). Todavia, apesar da proposta apresentar resultados satisfatórios ela não levou em consideração métricas importantes na avaliação de protocolos de roteamento para redes *wireless*, por exemplo, *overhead* de roteamento e *consumo de energia*, que podem influenciar no desempenho da rede principalmente se os nós forem móveis e não possuírem fontes de energia fixa (ex., tomada). Outra questão importante com relação a essa proposta é que os autores não apresentam as curvas do conjunto *Fuzzy* solução (ex., *Fuzzy Link Cost*) e não mencionam o método de composição das regras.

3. Descrição da Proposta

Na maioria dos casos, os protocolos de roteamento implementados em *MANETs* objetivam melhorar o desempenho das mesmas utilizando a métrica de roteamento com um único parâmetro de escolha de rotas como atraso fim-a-fim, número de saltos, taxa de perda de pacotes, vazão, etc [20]. Sabe-se que estes parâmetros de roteamento, em muitos casos, não são independentes. Por exemplo, métricas de roteamento que priorizam apenas a vazão podem levar a caminhos com altas perdas de pacotes [15]. Neste contexto, métricas de roteamento que possuem um único parâmetro, principalmente em

⁶Esta métrica foi aplicada ao *OLSR* nos seguintes trabalhos [15, 16].

MANETs, podem diminuir o desempenho da rede ao invés de melhorá-lo. Também, há o problema dos nós egoístas nas *MANETs* [9] e muitos pesquisadores apontam o comportamento egoísta como uma das principais causas de descarte de pacotes com o objetivo de economizar energia [6, 8]. Outra questão importante consiste no fato da detecção do nó egoísta, isto é, diferenciar o descarte de pacotes por egoísmo (intencional) de um descarte de pacotes por eventos ocorridos nas camadas de rede e enlace de dados não é uma tarefa trivial [18]. Portanto, um protocolo de roteamento eficiente deve possuir uma métrica de roteamento que considere múltiplos parâmetros. Por exemplo, métricas de roteamento que levam em consideração a energia residual e a taxa de perda de pacotes podem levar um nó a escolher rotas sem nós egoístas, uma vez que nós com pouca energia tendem a ser egoístas [5].

A métrica proposta neste artigo foi projetada para melhorar a qualidade do tráfego e minimizar o impacto dos nós egoístas nas *MANETs* utilizando três parâmetros para seleção de rotas: (i) *Índice de Perda de Pacotes (IPP)*, que deve ser minimizado por ser um dos principais indícios do comportamento egoísta [18]; (ii) *Energia Residual (ER)*⁷, este parâmetro deve ser maximizado por ser um dos recursos mais escassos em *MANETs* [8]; e (iii) *Índice de Conectividade (IC)*⁸, neste caso deve ser maximizado, pois ele é um dos principais responsáveis pelo desempenho das *MANETs* [3]. Assim, cada nó na rede difundirá o custo de redirecionar um pacote com base na seguinte função, Equação 1:

$$FC = f(IPP, ER, IC). \quad (1)$$

3.1 Sistema de Inferência Fuzzy Proposto

Um *Sistema de Inferência Fuzzy (SIF)* (Figura 1) é baseado na teoria dos conjuntos *Fuzzy* [11] e é formado por três partes distintas: *Fuzzificação*, *Inferência* e *Defuzzificação*. Cada uma dessas partes é responsável pelas seguintes ações:

- *Fuzzificação*: mapeia um valor de entrada não *Fuzzy* (ex., valor *crisp*, os parâmetros *IPP*, *ER* e *IC*) em um valor *Fuzzy*;
- *Inferência*: mapeia um valor *Fuzzy* em outro. A *Inferência* valida regras, denominadas *Regras de Inferência*, que são sentenças ***IF P is Good AND B is Medium Then C is Good***, em que ***IF P is Good AND B is Medium*** é o antecedente da regra; ***Then C is Good*** é o consequente; ***P, B e C*** são *Variáveis Linguísticas (VLs)*⁹; ***Good*** e ***Medium*** são conceitos linguísticos, conjuntos *Fuzzy*;
- *Defuzzificação*: converte um número *Fuzzy* para uma quantidade precisa (*crisp* ou real), que possa ser usada

na tomada de decisão (ex., *Custo Fuzzy (FC)*, Equação 1).

O *SIF* descrito neste artigo é baseado no modelo de inferência de Mamdani [22] e as entradas a serem fuzzificadas são as *VLs IPP, ER* e *IC* que representam as perdas de pacotes, a carga da bateria e a conectividade do nó que executa o cálculo. A *VL* de saída do sistema consiste no *Custo Fuzzy (FC)*, o custo de encaminhamento que será difundido na rede pelo módulo de envio do *OLSR*, ou seja, o *SIF* proposto tem como entrada as *VLs IPP, ER* e *IC* e como saída a *VL Custo* (ex., *Custo Fuzzy* ou métrica de roteamento *FC*). Os conceitos linguísticos *Low*, *Medium* e *High*, são usados para descrever a qualidade que cada *VL* pode possuir em um ambiente *MANET* e designam o grau de pertinência que cada *VL* pode assumir nas funções de pertinência. As funções de pertinência, apresentadas nas Figuras 2, 3, 4 e 5 representam estas *VLs*.

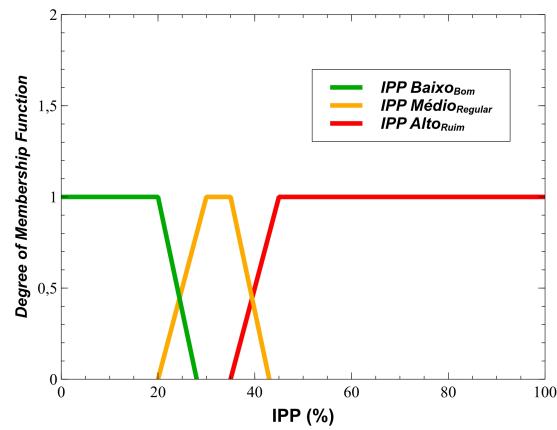


Figura 2. Função de Pertinência de Entrada (Antecedente), Variável Linguística *IPP*.

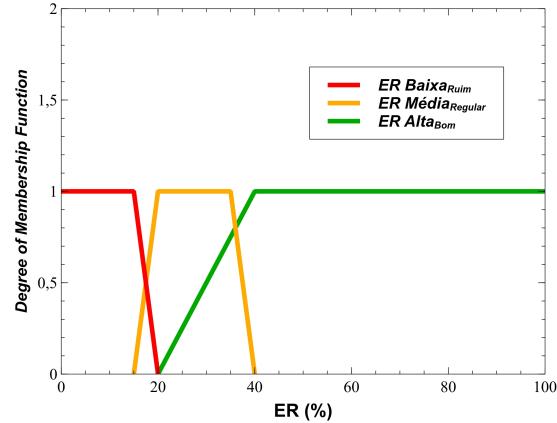


Figura 3. Função de Pertinência de Entrada (Antecedente), Variável Linguística *ER*.

⁷*ER*: consiste na energia que um nó possui em um instante de tempo.

⁸*IC*: consiste na quantidade de vizinhos a um salto de distância que um nó possui em um instante de tempo.

⁹*VLs*: são os espaços dos conjuntos *Fuzzy* nas funções de pertinência. E são usadas para a criação de regras que representam o conhecimento humano sobre o sistema a ser modelado.

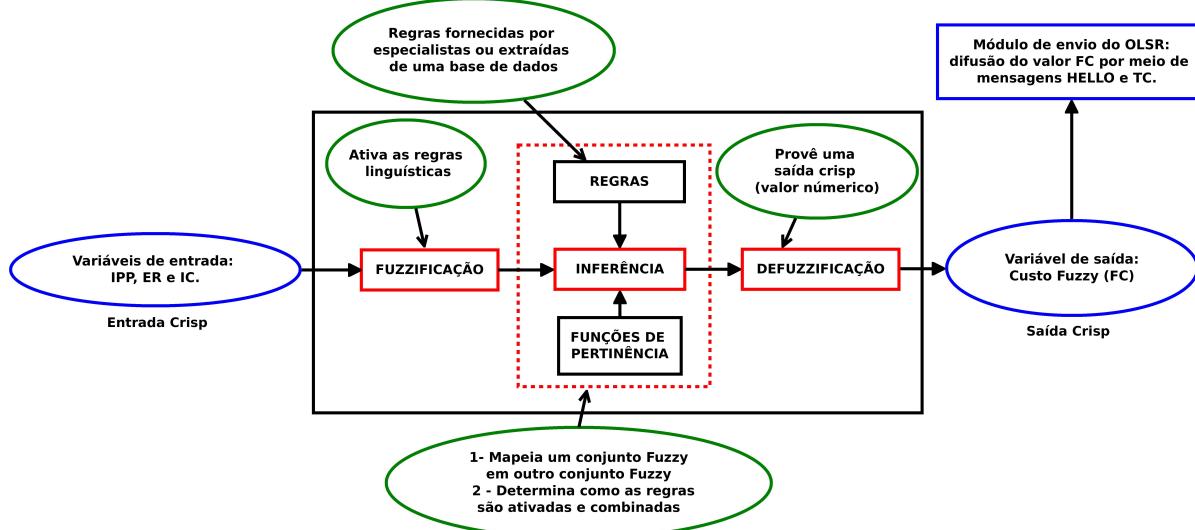


Figura 1. Sistema de Inferência Fuzzy. Baseado em [21].

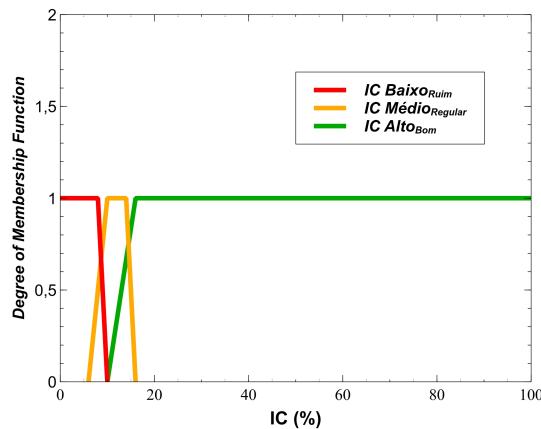


Figura 4. Função de Pertinência de Entrada (Antecedente), Variável Linguística *IC*.

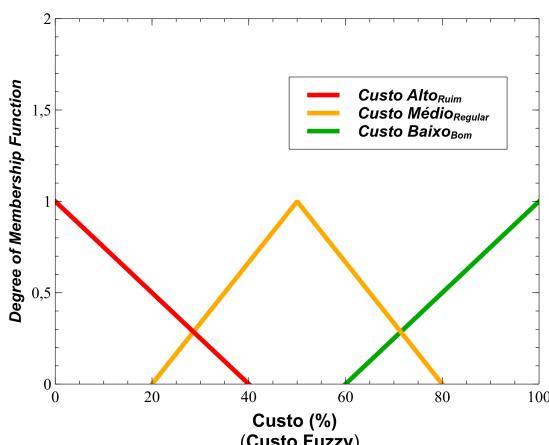


Figura 5. Função de Pertinência de Saída (Consequente), Variável Linguística *Custo*.

Estas funções de pertinência foram definidas com base no conhecimento de especialistas em *MANETs*. Já os tipos de curvas utilizadas na construção das funções de pertinência foram trapezoidais e triangulares, isso se deve ao fato desses tipos de curvas, ao contrário das curvas senoidais e gaussianas, representarem melhor a natureza do problema a ser tratado e também por serem as mais eficientes computacionalmente [23], veja as Equações 2 (ex., curva trapezoidal) e 3 (ex., curva triangular).

$$\mu_A(x) = \text{MAX} \left(\text{MIN} \left(\frac{(x-a)}{(b-a)}, 1, \frac{(d-x)}{(d-c)} \right), 0 \right). \quad (2)$$

$$\mu_A(x) = \text{MAX} \left(\text{MIN} \left(\frac{(x-a)}{(b-a)}, \frac{(c-x)}{(c-b)} \right), 0 \right). \quad (3)$$

As regras implementadas no *SIF* são do tipo **IF A AND B THEN C**, todas com operador lógico **AND**. Esse operador é requerido toda vez que é necessário considerar todos os predicados de uma regra. Por exemplo, múltiplos parâmetros no processo de escolha de rotas (ex., *IPP*, *ER* e *IC*). Uma operação *Fuzzy AND* (ex., intersecção), também denominada *T-norm*, de dois conjuntos *Fuzzy* *A* e *B* com as respectivas funções membros $\mu_A(x)$ e $\mu_B(x)$, é um conjunto *Fuzzy* *C* = $A \cap B$, que pode ser representado de duas formas (Equações 4 e 5):

$$\mu_C(x) = \text{MIN}(\mu_A(x), \mu_B(x)), x \in X. \quad (4)$$

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x). \quad (5)$$

Neste contexto, foram criadas 27 regras de inferência, pois são três *VLS* de entrada *IPP*, *ER* e *IC* que podem possuir três diferentes conceitos linguísticos *Low*, *Medium* ou *High*, logo considerando a relação entre as *VLS* e os conceitos linguísticos a quantidade total das regras pode ser obtida por meio de um

Arranjo com Repetição A, $R_{n,p} = |n|^{|p|}$ com $n = \{IPP, ER, IC\}$ e $p = \{Low, Medium, High\}$. Dessa forma, o total de regras é dado por $A, R_{3,3} = 3^3 = 27$. Entretanto, como um dos objetivos da proposta consiste em escolher rotas com baixas taxas de perdas de pacotes para evitar nós egoístas, todas as regras que possuem a VL *IPP* com conceito linguístico *High* foram mescladas em uma única regra: ***IF IPP is High \Rightarrow Fuzzy Cost is High***. Em função disso, foram selecionadas 19 regras de inferência das 27 possíveis. Cada regra tem um consequente (*Custo Fuzzy*), que pode implicar em *Low*, *Medium* ou *High*. As 19 regras do *SIF* proposto são apresentadas na Tabela 1.

A operação entre os predicados das regras (P_i^k)¹⁰ é realizada com base no significado dos conceitos linguísticos dos parâmetros *IPP*, *ER* e *IC*. Devido a isso, os consequentes das regras de inferência são gerados, baseando-se nos seguintes critérios: relevância dos conceitos linguísticos e operações lógicas aplicadas aos qualificadores desses conceitos em cada predicado. Essa relevância (ex., qualidade) ou significado é apresentada na Tabela 2.

Tabela 2. Relevância dos Conceitos Linguísticos das VLs e seu Significado.

Variável Linguística	Conceitos Linguísticos	Significado/Qualificador
<i>IPP</i>	<i>Low</i>	<i>Good</i>
<i>IPP</i>	<i>Medium</i>	<i>Moderate</i>
<i>IPP</i>	<i>High</i>	<i>Bad</i>
<i>ER</i>	<i>Low</i>	<i>Bad</i>
<i>ER</i>	<i>Medium</i>	<i>Moderate</i>
<i>ER</i>	<i>High</i>	<i>Good</i>
<i>IC</i>	<i>Low</i>	<i>Bad</i>
<i>IC</i>	<i>Medium</i>	<i>Moderate</i>
<i>IC</i>	<i>High</i>	<i>Good</i>
<i>Custo</i>	<i>Low</i>	<i>Good</i>
<i>Custo</i>	<i>Medium</i>	<i>Moderate</i>
<i>Custo</i>	<i>High</i>	<i>Bad</i>

A conclusão de cada regra, exceto as regras **1**, **13** e **15**, segue o algoritmo de *Implicação Fuzzy (IF)*:

1. Se todas as regras possuírem conceitos linguísticos diferentes (ex., *Low*, *Medium*, *High*) ou iguais (ex., *Low*, *Low*, *Low*), para a conclusão de tal regra, aplica-se o operador *AND* de forma sequencial ao qualificador da regra (ex., *Good*, *Moderate* e *Bad*). Por exemplo, a regra **8** (Tabela 1) ***IF IPP is Medium, AND Residual Energy is Low, AND Connectivity is High***. Nesse contexto, a conclusão dessa regra será ***Fuzzy Cost is Medium***;
2. Se uma regra tiver duas VLs com conceitos linguísticos iguais (ex., *Medium* e *Medium*), para a conclusão de tal regra aplica-se o operador *AND* de forma sequencial somente a essas VLs. Por exemplo, a regra **5** (Tabela 1) ***IF IPP is Medium, AND Residual Energy is Medium, AND Connectivity is High***. Em função dessa situação, o consequente dessa regra será ***Fuzzy Cost is Medium***, e não ***Fuzzy Cost is Low***.

¹⁰ Predicado i da regra k . Exemplo: $P_2^7 = \text{Residual Energy is Medium}$, Tabela 1.

O algoritmo *IF* foi elaborado de modo a impedir que apenas um predicado influencie na conclusão da regra. As operações realizadas com os qualificadores para conclusão das regras estão descritas na Tabela 3. As exceções ao algoritmo *IF* consistem nas regras **1**, **13** e **15**. O algoritmo *IF* não se aplica à regra **1** pelo fato da mesma ter apenas um predicado. Já a não aplicação do algoritmo *IF* às regras **13** e **15** ocorre porque essas regras não possuem nenhum qualificador *Bad*¹¹ para *IPP* e *ER*. Portanto, mesmo tendo conectividade baixa/regular há uma alta probabilidade do nó entregar os pacotes. Isso significa que elas não levarão o nó a escolher rotas compostas por nós egoístas e com baixa capacidade energética. Em função disso, a conclusão dessas regras, **13** e **15**, será *Low* ao invés de *Medium*.

Tabela 3. Operações Realizadas Sobre os Qualificadores das Regras de Inferência.

Qualificador	Operador	Qualificador	Implicação	Resultado
<i>Good</i>	<i>AND</i>	<i>Good</i>	\Rightarrow	<i>Good</i>
<i>Good</i>	<i>AND</i>	<i>Moderate</i>	\Rightarrow	<i>Good</i>
<i>Good</i>	<i>AND</i>	<i>Bad</i>	\Rightarrow	<i>Moderate</i>
<i>Moderate</i>	<i>AND</i>	<i>Moderate</i>	\Rightarrow	<i>Moderate</i>
<i>Moderate</i>	<i>AND</i>	<i>Bad</i>	\Rightarrow	<i>Bad</i>
<i>Bad</i>	<i>AND</i>	<i>Bad</i>	\Rightarrow	<i>Bad</i>

Há diversos métodos de composição descritos na literatura, porém o método utilizado neste trabalho é o *MAX/MIN* (conforme Equação 6) [24], característico do modelo de inferência de Mamdani. A composição de regras é derivada da agregação de conjuntos *Fuzzy* produzidos pela interação de várias sentenças, cujo objetivo é gerar um valor de saída que será utilizado na defuzzificação.

$$\mu_{B^i}(y) = \text{MAX}(\text{MIN}[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \dots, \mu_{A_p^k}(x_p)]_{\mu_{B_i}(y)}), \quad \forall y \in U_{yj}. \quad (6)$$

Em que:

- k : é o número da regra, com $k = 1, \dots, n$;
- p : é o predicado da regra k , com $p = 1, \dots, n$.
- $\mu_{B^i}(y)$: função de pertinência do conjunto *Fuzzy* solução;
- $\mu_{B_i}(y)$: função de pertinência do conjunto *Fuzzy* de saída da regra k ;
- U_{yj} : universo de discurso do conjunto *Fuzzy* de saída.

O método de defuzzificação utilizado foi a média ponderada, pois ele é um dos mais eficientes computacionalmente, de acordo com a Equação 7. Não existe um procedimento único para a escolha do método de defuzzificação [22].

$$z^* = \frac{\sum \mu_A \bar{x} \cdot \bar{x}}{\sum \mu_A \bar{x}}. \quad (7)$$

¹¹ Para sanar dúvidas sobre qualificadores revise a Tabela 2.

Tabela 1. Regras de Inferência.

N°	Regras
1*	<i>IF IPP is High \Rightarrow Fuzzy Cost is High</i>
2	<i>IF IPP is Medium, AND Residual Energy is High AND Connectivity is High \Rightarrow Fuzzy Cost is Low</i>
3	<i>IF IPP is Medium, AND Residual Energy is High AND Connectivity is Medium \Rightarrow Fuzzy Cost is Medium</i>
4	<i>IF IPP is Medium, AND Residual Energy is High, AND Connectivity is Low \Rightarrow Fuzzy Cost is Medium</i>
5	<i>IF IPP is Medium, AND Residual Energy is Medium, AND Connectivity is High \Rightarrow Fuzzy Cost is Medium</i>
6	<i>IF IPP is Medium, AND Residual Energy is Medium AND Connectivity is Medium \Rightarrow Fuzzy Cost is Medium</i>
7	<i>IF IPP is Medium, AND Residual Energy is Medium, AND Connectivity is Low \Rightarrow Fuzzy Cost is Medium</i>
8	<i>IF IPP is Medium, AND Residual Energy is Low, AND Connectivity is High \Rightarrow Fuzzy Cost is Medium</i>
9	<i>IF IPP is Medium, AND Residual Energy is Low, AND Connectivity is Medium \Rightarrow Fuzzy Cost is Medium</i>
10	<i>IF IPP is Medium, AND Residual Energy is Low, AND Connectivity is Low \Rightarrow Fuzzy Cost is High</i>
11	<i>IF IPP is Low, AND Residual Energy is High , AND Connectivity is High \Rightarrow Fuzzy Cost is Low</i>
12	<i>IF IPP is Low, AND Residual Energy is High, AND Connectivity is Medium \Rightarrow Fuzzy Cost is Low</i>
13*	<i>IF IPP is Low, AND Residual Energy is High, AND Connectivity is Low \Rightarrow Fuzzy Cost is Low</i>
14	<i>IF IPP is Low, AND Residual Energy is Medium, AND Connectivity is High \Rightarrow Fuzzy Cost is Low</i>
15*	<i>IF IPP is Low, AND Residual Energy is Medium , AND Connectivity is Medium \Rightarrow Fuzzy Cost is Low</i>
16	<i>IF IPP is Low, AND Residual Energy is Medium, AND Connectivity is Low \Rightarrow Fuzzy Cost is Medium</i>
17	<i>IF IPP is Low, AND Residual Energy is Low, AND Connectivity is High \Rightarrow Fuzzy Cost is Medium</i>
18	<i>IF IPP is Low, AND Residual Energy is Low, AND Connectivity is Medium \Rightarrow Fuzzy Cost is Medium</i>
19	<i>IF IPP is Low, AND Residual Energy is Low, AND Connectivity is Low \Rightarrow Fuzzy Cost is High</i>

* Consistem em regras que não seguem o algoritmo de *Implicação Fuzzy (IF)*

O valor de μ_{Ax} é obtido da composição e z^* é o custo de encaminhamento de pacotes do nó que executa o cálculo (FC), conforme Equação 1.

3.2 Funcionamento do OLSR-FC

No *OLSR-FC* cada nó calcula o *Custo Fuzzy* localmente, com base em suas informações locais, por exemplo, a taxa de perda de pacotes (*IPP*), a energia residual (*ER*) e o índice de conectividade (*IC*), de acordo com a Equação 1. Dessa forma, após a computação do *Custo Fuzzy*, o nó que executa o cálculo o difunde por meio das mensagens *HELLO* e *TC* modificadas, denominadas *FUZZY_HELLO* e *FUZZY_TC*, estas possuem dois campos adicionais: *Fuzzy Cost* e *Neighbor Fuzzy Cost*, que transportam o *Custo Fuzzy* do nó que executa o cálculo e o *Custo Fuzzy* de todos os seus vizinhos. Nesse contexto, observa-se que a frequência de captura dos parâmetros *IPP*, *ER* e *IC*, e a execução do *SIF* dependem da frequência de envio das mensagens *FUZZY_HELLO* e *FUZZY_TC*. Dessa forma, este processo é realizado toda vez que um nó precisa enviar uma mensagem *FUZZY_HELLO* e/ou *FUZZY_TC*. Com relação ao custo computacional da proposta não há cálculos complexos, pois a mesma utiliza-se de curvas e método de defuzzificação com baixo custo computacional [23, 22], logo o maior custo está na execução do algoritmo de *Dijkstra* para selecionar a rota com o menor *Custo Fuzzy*, $O(|V|^2)$ [25]. Portanto, o *OLSR-FC* pode ser executado em dispositivos com poucos recursos de *hardware*.

3.2.1 Mensagem FUZZY_HELLO

Mensagens *FUZZY_HELLO* são difundidas periodicamente a cada dois segundos para seus vizinhos a um salto, da mesma forma que a mensagem *HELLO* no *OLSR RFC 3626*. Dessa maneira, o nó receptor da mensagem passará a ter o conhecimento do custo de encaminhamento de pacotes do nó transmissor, campo *Fuzzy Cost*, e de todos os seus vizinhos, campo *Neighbor Fuzzy Cost*, de quem o mesmo recebeu mensagens

FUZZY_HELLO previamente. O formato dessa mensagem é apresentado na Figura 6.

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
|       Reserved      |     Htime    | Willingness |
+-----+-----+-----+-----+-----+-----+-----+
|   Link Code   |   Reserved   | Link Message Size
+-----+-----+-----+-----+-----+-----+-----+
|               Neighbor Interface Address
+-----+-----+-----+-----+-----+-----+-----+
|       Fuzzy Cost   |   Neighbor   | Fuzzy Cost
+-----+-----+-----+-----+-----+-----+-----+
|               Neighbor Interface Address
+-----+-----+-----+-----+-----+-----+-----+
|       Fuzzy Cost   |   Neighbor   | Fuzzy Cost
+-----+-----+-----+-----+-----+-----+-----+
|               Neighbor Interface Address
+-----+-----+-----+-----+-----+-----+-----+
|       .          |   .          | .
+-----+-----+-----+-----+-----+-----+-----+

```

Figura 6. Mensagem *FUZZY_HELLO* com Informações do Custo Fuzzy.

3.2.2 Mensagem FUZZY_TC

A mensagem *FUZZY_TC* difunde o *Custo Fuzzy* do nó transmissor, campo *Fuzzy Cost*, e de todos os nós vizinhos anunciados na mesma, campo *Neighbor Fuzzy Cost*. Assim como uma mensagem *TC* no *OLSR RFC 3626*, a mensagem *FUZZY_TC* transporta informações sobre a topologia da rede para ativar o roteamento, e a mesma é enviada por um nó a cada cinco segundos para seus vizinhos a um salto; todos os receptores processam as mensagens *FUZZY_TC*, mas apenas a encaminham para os nós que estão a dois saltos do nó enviaador se o mesmo estiver no seu conjunto de seletores de *MPRs*. O formato da mensagem *FUZZY_TC* é apresentado na Figura 7.

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1	2 3 4 5 6 7 8 9 0 1	2 3 4 5 6 7 8 9 0 1	2 3 4 5 6 7 8 9 0 1
ANSN	Reserved		
Advertised Neighbor Main Address			
Fuzzy Cost	Neighbor Fuzzy Cost		
Advertised Neighbor Main Address			
Fuzzy Cost	Neighbor Fuzzy Cost		
...			

Figura 7. Mensagem FUZZY_TC com Informações do Custo Fuzzy.

Nesta proposta, um nó após ter o conhecimento da topologia da rede, que foi sinalizada previamente por meio de mensagens FUZZY_TC recebidas, construirá um grafo ponderado com os custos de encaminhamento (ex., *Custo Fuzzy*) e executará o algoritmo de *Dijkstra*, em detrimento do algoritmo padrão de roteamento do OLSR, para criar sua tabela de roteamento. Portanto, a seleção de rotas entre o nó atual e um nó destino terá como critério o menor *Custo Fuzzy (Low)*, isto é, a menor somatória dos *Custos Fuzzy* de todos os enlaces que fazem parte da rota. No caso de haver duas rotas, ou mais, com o mesmo custo de encaminhamento será selecionada a que tiver o menor número de saltos. A Figura 8 apresenta o diagrama de estados do funcionamento do OLSR-FC.

3.3 Teste do SIF

Com o objetivo de avaliar o funcionamento do *SIF* foi desenvolvido um simulador para testar os processos de *Fuzzificação*, *Inferência*, *Composição* e *Defuzzificação*. Neste contexto, foram feitos exaustivos testes com diferentes valores para os parâmetros *IPP*, *ER* e *IC*. Além disso, foi realizado o cálculo do *Custo Fuzzy* de forma manual e o resultado foi comparado ao resultado do simulador.

Para realização do teste de avaliação foi utilizado o cenário com 50 nós, o mesmo usado para gerar os resultados da Seção 5. Os valores de entrada para as *VLs* antecedentes são: *IPP* = 15,2222; *ER* = 25,7777; e *IC* = 8,9999.

Com esses valores o *SIF* ativa as seguintes regras 12, 13, 15 e 16, Tabela 1. Essas regras são ativadas porque o valor *x* do *IPP* tem grau de pertinência¹² 1 no conjunto *Fuzzy Baixo*, Figura 2. Da mesma forma, o valor *x* do parâmetro *ER* corta as curvas dos conjuntos *Fuzzy "Médio"* com pertinência = 1,000 e "Alto" com pertinência = 0,289, Figura 3. E por fim, o valor do parâmetro *IC* tem grau de pertinência 0,500 no conjunto *Fuzzy "Alto"* e 0,750 no conjunto *Fuzzy Médio*, Figura 4.

Os valores de pertinência supracitados dos parâmetros *IPP*, *ER* e *IC* geram custos para cada uma das regras mencionadas e os mesmos podem ser observados na Tabela 4. Os custos de cada regra são obtidos com a aplicação do algoritmo *IF* apresentado na Subseção 3.1.

¹²Por exemplo: $f(x)$ ou $\mu_A(x)$. Veja a Subseção 3.1.

Tabela 4. Regras de Inferência Ativas.

Nº	Regras Ativadas
12	<i>IF</i> 1,000 _{IPP} is Low AND 0,289 _{ER} is High AND 0,750 _{IC} is Medium ⇒ <i>Fuzzy</i> 0,289 _{Cost} is <i>Low</i>
13	<i>IF</i> 1,000 _{IPP} is Low AND 0,289 _{ER} is High AND 0,500 _{IC} is Low ⇒ <i>Fuzzy</i> 0,289 _{Cost} is <i>Low</i>
15	<i>IF</i> 1,000 _{IPP} is Low AND 1,000 _{ER} is Medium AND 0,750 _{IC} is Medium ⇒ <i>Fuzzy</i> 0,750 _{Cost} is <i>Low</i>
16	<i>IF</i> 1,000 _{IPP} is Low AND 1,000 _{ER} is Medium AND 0,500 _{IC} is Low ⇒ <i>Fuzzy</i> 0,500 _{Cost} is <i>Medium</i>

Os valores consequentes de cada regra são obtidos do mínimo da pertinência de cada *VL* e o valor do conjunto *Fuzzy* solução é obtido do máximo dos consequentes de mesmo conjunto *Fuzzy*. Este método é denominado *MAX/MIN*. Dessa forma, ao aplicar a Equação 6(MIN) têm-se os seguintes resultados,

$$R^{12} \rightarrow IPP = 1,000, ER = 0,289 \text{ e } IC = 0,750 \text{ então,}$$

$$R^{12} = MIN[1,000; 0,289; 0,750] = 0,289$$

$$R^{13} \rightarrow IPP = 1,000, ER = 0,289 \text{ e } IC = 0,500 \text{ então,}$$

$$R^{13} = MIN[1,000; 0,289; 0,500] = 0,289$$

$$R^{15} \rightarrow IPP = 1,000, ER = 1,000 \text{ e } IC = 0,750 \text{ então,}$$

$$R^{15} = MIN[1,000; 1,000; 0,750] = 0,750.$$

$$R^{16} \rightarrow IPP = 1,000, ER = 1,000 \text{ e } IC = 0,500 \text{ então,}$$

$$R^{16} = MIN[1,000; 1,000; 0,500] = 0,500.$$

Em seguida, aplica-se a Equação 6(MAX) para consequentes iguais, por exemplo:

- Consequente *Low*:

$$R^{12} = 0,289, R^{13} = 0,289 \text{ e } R^{15} = 0,750 \text{ então}$$

$$\mu_{Low \cdot 1}(y) = MAX[0,289; 0,289; 0,750] = 0,750.$$

- Consequente *Medium*:

$$R^{16} = 0,500 \text{ então}$$

$$\mu_{Medium \cdot 2}(y) = MAX[0,500] = 0,500.$$

Dessa forma, como resultado da composição, têm-se como saídas os conjuntos *Fuzzy Low* e *Medium* com os seguintes graus de pertinências 0,750_{Low} e 0,500_{Medium}. Essas saídas são projetadas na curva *Custo* (ex., Figura 5) como mostra a Figura 9.

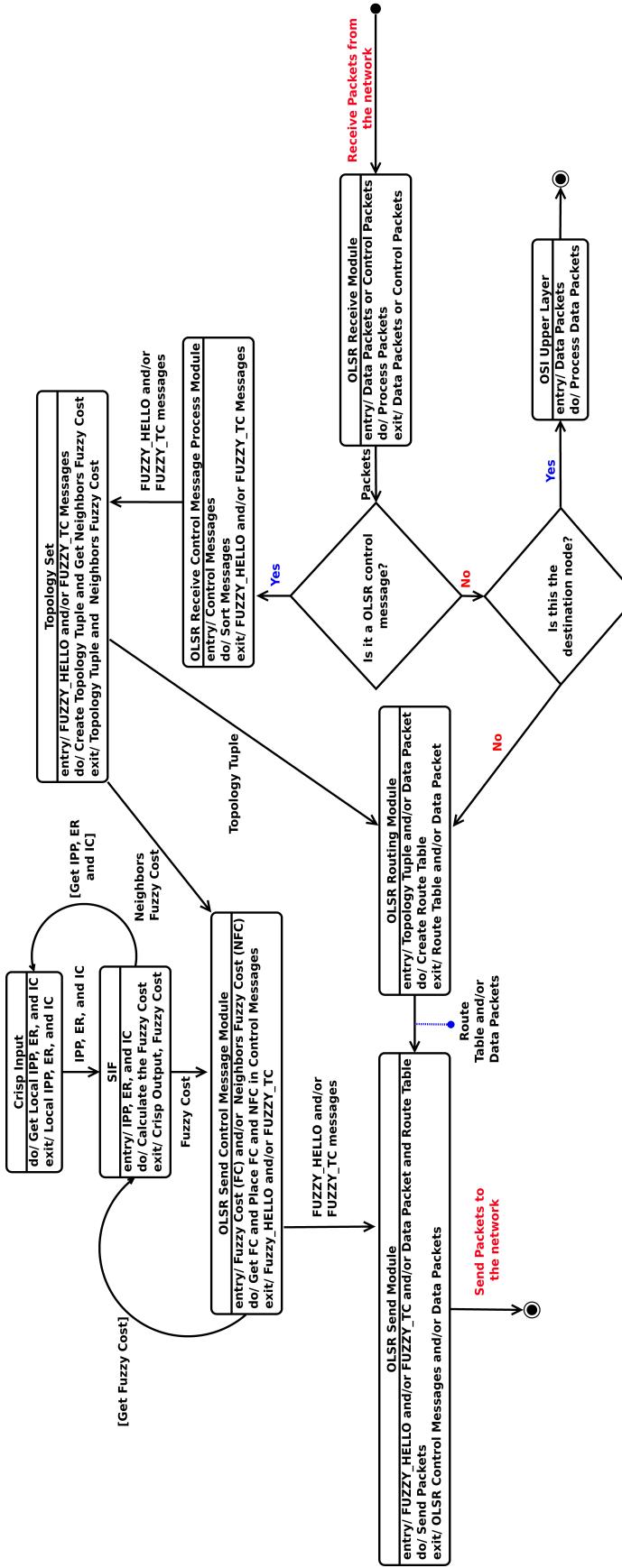


Figura 8. Diagrama de Estados do OLSR-FC.

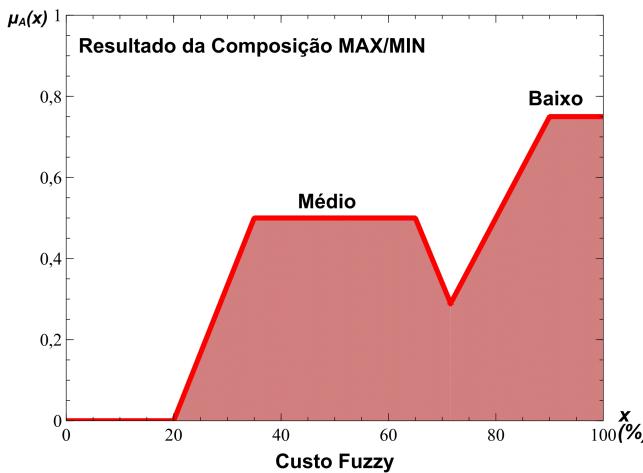


Figura 9. Conjunto Fuzzy Solução Gerado a Partir da Fuzzificação e Inferência.

A partir da composição encontra-se o conjunto Fuzzy solução e nesse conjunto aplica-se o método da *Média Ponderada* para realizar a *Defuzzificação*, de acordo com a Equação 7. Assim, o *Custo Fuzzy* é obtido como segue,

$$\begin{aligned} z^* &= \left(\frac{35,000 + 65,000}{2} \right) * 0,500 + \left(\frac{90,000 + 100,000}{2} \right) * 0,750 \\ &= 0,500 + 0,750 \\ &\Rightarrow z^* = 77. \end{aligned}$$

Já com a utilização do simulador, tem-se o resultado $z^* = 77$ conforme mostrado na Figura 10.

```
# ./main 15.2222 25.7777 8.9999 3 1 1
1 - Fuzzificação --> Converte valores reais não Fuzzy (IPP, ER e IC)
em valores Fuzzy:
Função IPP: u(15.2222)Bx 1.000, u(15.2222)Me 0.000 e u(15.2222)Al: 0.000
Função ER: u(25.778)Bx 0.000, u(25.778)Me 1.000 e u(25.778)Al: 0.289
Função IC: u(9.000)Bx 0.500, u(9.000)Me 0.750 e u(9.000)Al: 0.000
2 - Inferência --> Converte um Valor Fuzzy em outro valor Fuzzy:
2.1 - Regras com a composição MAX-MIN
R12: 0.289 - Custo Baixo (Bom)
R13: 0.289 - Custo Baixo (Bom)
R15: 0.750 - Custo Baixo (Bom)
R16: 0.500 - Custo Médio (Regular)
Max-Min Al: 0.000
Max-Min Me: 0.500
Max-Min Bx: 0.750
3 - Defuzzificação (Média Ponderada) + Max-Min: 77.00
Hardware e Software Utilizados:
```

Figura 10. Custo Fuzzy Obtido Através do Simulador.

De acordo com a Figura 10, os parâmetros passados ao simulador significam respectivamente: *<IPP ER IC>* *<Número de nós móveis>* *<Composição>* *<Defuzzificação>*, por exemplo:

- *<IPP ER IC>*: variáveis linguísticas que assumem os seguintes valores [0%, 100%];
- *<Número de nós móveis>*: dez (0), vinte (1), trinta (2) e cinquenta (3);
- *<Composição>*: *MAX/MIN* (1) e *MAX/PROD* (2);

- *<Defuzzificação>*: média ponderada (1), centroide discreta (2) e centroide contínua (3).

Como pode ser visto na Figura 10, o simulador proposto apresenta os resultados corretos para composição das regras, a ativação das regras e o processo de *Defuzzificação*. Dessa forma, para concluir o teste, é apresentado na Figura 11 o valor de z^* . E este resultado, que é gerado a partir da Figura 5, é o *Custo Fuzzy*.

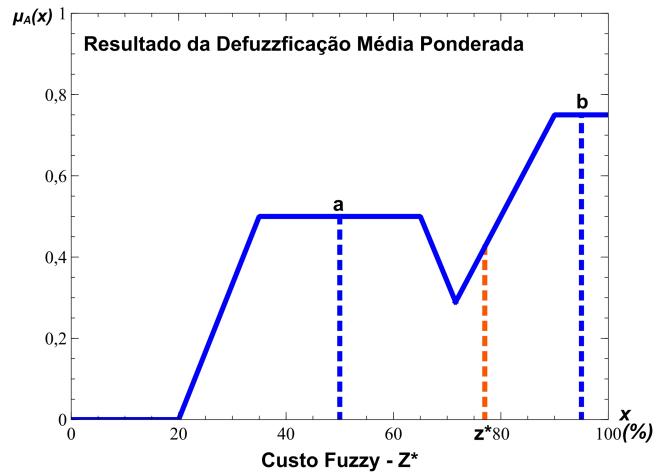


Figura 11. Representação Gráfica da Saída "crisp", o *Custo Fuzzy*.

Todos os processos do *SIF* (ex., *Fuzzificação*, *Inferência*, *Composição* e *Defuzzificação*), demonstrados nesta seção, dependem dos parâmetros de entrada *IPP*, *ER* e *IC*, portanto o resultado produzido pelo *SIF* pode ser representado como uma função, de acordo com a Equação 1.

O *Sistema de Inferência Fuzzy (SIF)* proposto foi desenvolvido em *C++* e implementa o mesmo código utilizado no simulador previamente descrito. Dessa forma, para avaliá-lo, antes da implementação no simulador *NS-2*, o mesmo foi modelado nas ferramentas *MATLAB 7.0 Fuzzy Tool Box* e *GNU Octave 4.2.2 fuzzy-logic-toolkit-0.4.5*, Figuras 12 e 13. A avaliação consistiu na inserção de 200 entradas, geradas aleatoriamente, dos parâmetros *IPP*, *ER* e *IC*, para verificar a precisão da nossa proposta quando comparada ao *MATLAB* e ao *GNU Octave*. Para propósitos de simplificação a Tabela 5 apresenta apenas os 20 primeiros resultados, observa-se que a nossa implementação possui resultados mais precisos que o *GNU Octave*, levando em consideração o *MATLAB* como referência de precisão. O teste apresentado utilizou-se do método de composição *MAX/MIN* e do método de defuzzificação da centroide porque as ferramentas *MATLAB* e *Octave* só utilizam o método da média ponderada apenas no modelo de inferência de *Takagi-Sugeno*. Portanto, como o método da média ponderada possui menor custo computacional, o mesmo foi implementado para o modelo de inferência de *Mamdani* como em [26].

```
[System]
Name='fuzzy50'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=19
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='IPP'
Range=[0 100]
NumMFs=3
MF1='Baixo':'trapmf',[0 0 20 28]
MF2='Medio':'trapmf',[20 30 35 43]
MF3='Alto':'trapmf',[35 45 100 100]

[Input2]
Name='ER'
Range=[0 100]
NumMFs=3
MF1='Baixo':'trapmf',[0 0 15 20]
MF2='Medio':'trapmf',[15 20 35 40]
MF3='Alto':'trapmf',[20 40 100 100]

[Input3]
Name='IC'
Range=[0 100]
NumMFs=3
MF1='Baixo':'trapmf',[0 0 8 10]
MF2='Medio':'trapmf',[6 10 14 16]
MF3='Alto':'trapmf',[10 16 100 100]

[Output1]
Name='Custo'
Range=[0 100]
NumMFs=3
MF1='Alto':'trimf',[0 0 40]
MF2='Medio':'trimf',[20 50 80]
MF3='Baixo':'trimf',[60 100 100]
```

Figura 12. Sistema de Inferência Fuzzy, Métrica de Roteamento Fuzzy Cost, Modelado no MATLAB.

Em ambos os *SIFs* mostrados nas Figuras 12 e 13, o conjunto de regras é codificado logo abaixo do código da VL de saída *Custo* (ex., **[Output1]**). Todavia, o mesmo foi omitido nessas figuras com propósito de simplificação. Por exem-

plho, o código de uma regra, no *MATLAB* e *GNU Octave*, equivalente à regra 1: **IF IPP is High \Rightarrow Fuzzy Cost is High** (Tabela 1) seria **3 0 0**, 1, em que **3** consiste no conceito linguístico da VL *IPP* (ex., *High*), o **0** significa que as duas VLs *ER/IC* não existem nessa regra e **1** é a conclusão da regra, o conceito linguístico da VL *Custo* (ex., *High*).

```
[System]
Name='fuzzy50'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=19
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='IPP'
Range=[0 100]
NumMFs=3
MF1 = 'Baixo' : 'trapmf', [-1 0 20 28]
MF2 = 'Medio' : 'trapmf', [20 30 35 43]
MF3 = 'Alto' : 'trapmf', [35 45 100 101]

[Input2]
Name='ER'
Range=[0 100]
NumMFs=3
MF1 = 'Baixo' : 'trapmf', [-1 0 15 20]
MF2 = 'Medio' : 'trapmf', [15 20 35 40]
MF3 = 'Alto' : 'trapmf', [20 40 100 101]

[Input3]
Name='IC'
Range=[0 100]
NumMFs=3
MF1 = 'Baixo' : 'trapmf', [-1 0 8 10]
MF2 = 'Medio' : 'trapmf', [6 10 14 16]
MF3 = 'Alto' : 'trapmf', [10 16 100 101]

[Output1]
Name = 'Custo'
Range = [0 100]
NumMFs = 3
MF1 = 'Alto' : 'trimf', [-1 0 40]
MF2 = 'Medio' : 'trimf', [20 50 80]
MF3 = 'Baixo' : 'trimf', [60 100 101]
```

Figura 13. Sistema de Inferência Fuzzy, Métrica de Roteamento Fuzzy Cost, Modelado no *GNU Octave*.

Tabela 5. Comparação do Sistema de Inferência Proposto (ex., métrica *FC*) com *MATLAB* e *GNU Octave*.

Parâmetros de Entrada (VLs)			Custos de Saída		
IPP	ER	IC	FC	OCTAVE	MATLAB
4,85851	48,57571	69,47471	87,00000	86,67500	87,00000
31,08081	58,61611	71,49491	87,00000	86,67500	87,00000
21,60601	7,25251	96,66661	50,00000	50,00000	50,00000
28,05051	5,19191	29,95951	50,00000	50,00000	50,00000
16,17171	94,65651	89,60601	87,00000	86,67500	87,00000
7,74741	3,57571	54,58581	50,00000	50,00000	50,00000
23,62621	18,62621	24,07071	66,32539	66,02700	66,32540
37,73731	70,95951	32,73731	61,90530	61,76500	61,91440
4,73731	37,07071	66,16161	86,71358	86,42400	86,71360
27,11111	5,18181	70,81811	50,00000	50,00000	50,00000
20,49491	57,76761	25,27271	86,93246	86,62500	86,93250
10,97971	39,19191	29,08081	86,96527	86,65200	86,96530
22,76761	15,03031	82,74741	50,19194	50,18600	50,19190
40,39391	74,83831	28,28281	42,91720	42,99700	42,91170
21,44441	97,46461	98,93931	86,58888	86,30500	86,58890
6,87871	47,44441	92,34341	87,00000	86,67500	87,00000
12,58581	15,49491	3,96961	20,59820	20,98800	20,59820
7,27271	85,90901	36,17171	87,00000	86,67500	87,00000
35,90901	66,00001	65,61611	76,71182	76,44100	76,73550
2,23231	78,20201	69,02021	87,00000	86,67500	87,00000

```
%MATLAB Fuzzy ToolBox Code:
% Read the FIS structure from a file.
fis = readfis('fuzzy50.fis');
% Calculate Fuzzy Cost:
fprintf ('\nFor the following values of metics (IPP, ER and IC):\n\n')
fprintf ('\nFuzzy Cost:\n\n')
fc = evalfis([15.2222 25.7777 8.9999], fis, 101)

##GNU Octave Fuzzy ToolBox Code:
## Read the FIS structure from a file.
fis = readfis ('fuzzy50');
## Demonstrate showrule with hedges.
## Calculate Fuzzy Cost:
puts ("\nFor the following values of metics (IPP, ER and IC):\n\n")
puts ("\nFuzzy Cost:\n\n");
fc = evalfis ([15.2222 25.7777 8.9999], fis, 101)
```

Figura 14. Código de Execução do *SIF* no *MATLAB* e *GNU Octave*. O Código Apresentado Aqui Chama os Arquivos Mostrados nas Figuras 12 e 13.

O *SIF* proposto nesta pesquisa também implementa o método de defuzzificação da centroide. Portanto, para avaliá-lo com este método, foram realizados testes no *MATLAB*, *GNU Octave* e no simulador, com as mesmas entradas previamente descritas nesta seção (ex., $IPP = 15,2222$; $ER = 25,7777$; e $IC = 8,9999$) conforme mostra a Figura 14. Os resultados dos testes mostram os seguintes valores da métrica *FC*:

- Métrica *FC* implementada no *MATLAB*: 66,3893;
- Métrica *FC* implementada no *GNU Octave*: 66,061;
- Métrica *FC* implementada em *C++*: 66,3893.

4. Materiais e Métodos

A metodologia empregada para avaliação do *OLSR-FC* foi determinada por:

1. **Simulador:** *NS-2* 2.34, versão modificada por Cordeiro *et al.* [16]¹³;

¹³Todas as métricas descritas na Seção 2.2 foram implementadas no *NS-2*, versão 2.34, por Cordeiro *et al.* e estas foram usadas na avaliação da métrica *MD*, proposta pelo mesmo.

2. **Modelo de Mobilidade:** cada nó foi distribuído aleatoriamente em uma área com as seguintes dimensões $1000m \times 1000m$ e o modelo de mobilidade usado foi o *Random Waypoint (RW)* [27], este padrão de movimentação tem sido a base para a maioria das avaliações de *MANETs*. A ferramenta usada para gerar a mobilidade dos nós foi o *setdest*, que acompanha o *NS-2* e está disponível no seguinte diretório */ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen/setdest/*;

3. **Quantidade de Fluxos de Transmissão:** foram usados 12 fluxos *CBR* a uma taxa de $128Kbps$ para que fosse possível, em um ambiente sem nós egoístas, atingir uma taxa de entrega de pacotes superior a 80%. Os fluxos foram iniciados entre 10 e 11 segundos de simulação. O objetivo disso consistiu em permitir que o protocolo de roteamento estabelecesse as rotas antes que pacotes de dados fossem enviados;

4. **Quantidade de Simulações:** os resultados foram obtidos da média aritmética de 10 simulações, e foram usadas sementes de distribuição aleatória e nível de confiança de 95%. As simulações foram feitas de forma sequencial, uma após a outra, de modo que uma simulação não interferisse no resultado da outra. Foram utilizados um cenário (*C*) com 50 nós; cinco ambientes (*A*) com as seguintes taxas de nós egoístas 0%, 10%, 20%, 30% e 40%; quatro métricas de roteamento (*M*) *FC*, *ML*, *ETX* e *MD*; e dez execuções de simulação (*S*). Dessa forma, a quantidade total de simulações (*N*) foi obtida como segue:

$$N = C * A * M * S \Rightarrow 1 * 5 * 4 * 10 = 200. \quad (8)$$

5. **Hardware e Software Utilizados:** processador *Intel(R) CORE(TM) i5 CPU M 450 2.40GHz*, 6GB de memória *RAM* e 500GB de *HD*, sistema operacional *Ubuntu Linux 12.04 LTS (Precise Pangolin) 64Bits*;

6. **Métricas de Desempenho:** as métricas de desempenho descritas aqui não têm nenhuma relação direta com os parâmetros de entrada do *SIF*: *IPP*, *ER* e *IC*. Elas apenas tratam-se de métricas utilizadas na avaliação comparada entre o que é proposto e as demais extensões do *OLSR*.

• **Taxa de Perda de Pacotes (TPP) -** é a quantia de pacotes gerados na fonte menos a quantia de pacotes recebidos no destino [28], Equação 9. Estas perdas podem ocorrer por vários motivos, por exemplo, rotas indisponíveis, *buffer* cheio, congestionamento, erros nas camadas, interferência, enlaces falhos, intencionalmente (ex., nós egoístas e/ou maliciosos);

$$TPP = \sum_{i=1}^n \text{PacotesEnviados} - \sum_{i=1}^n \text{PacotesRecebidos}. \quad (9)$$

- **Atraso Fim-a-Fim (AFF)** - refere-se ao tempo levado por um pacote desde sua geração na camada de aplicação do nó emissor até o seu recebimento na camada de aplicação do nó destino [28]. O atraso fim-a-fim (Equação 10) leva em consideração todos os atrasos de redes: atraso de propagação, atraso de transmissão, atraso de enfileiramento e atraso de processamento;

$$AFF = \text{Tempo}_{\text{Destino}}^{\text{Pacote}} - \text{Tempo}_{\text{Fonte}}^{\text{Pacote}}. \quad (10)$$

- **Jitter** - é definido como a variação do atraso e consiste em uma importante métrica de desempenho para aplicações de áudio e vídeo, Equação 11. Neste tipo de cálculo de *Jitter*, pois existem vários, mesmo que a fonte não esteja gerando pacotes a uma taxa constante ele não apresenta nenhum erro nas medições [29];

$$Jitter = |AFF_n - AFF_{n-1}|. \quad (11)$$

- **Consumo de Energia (CE)** - é a quantidade de energia utilizada por um nó para manutenção das rotas, recebimento e envio de pacotes [30]. O consumo pode ser medido em Joules (*J*) [13] ou Watts (*W*), Equação 12. Um nó ao ser desativado, por falta de energia, pode diminuir o tempo de vida da rede. Em função disso, muitas métricas de roteamento tem utilizado a energia residual como parâmetro de escolha de rotas [8];

$$CE = E_{\text{Inicial}} - E_{\text{Final}}. \quad (12)$$

E_{Inicial} é a capacidade da bateria do nó no início da simulação, por exemplo, 100W. E E_{Final} consiste na energia restante do nó após a simulação. CE não deve ser confundido com ER (ex., um dos parâmetros de entrada do *SIF*) que é medida sempre que um nó precisa enviar uma mensagem *FUZZY_HELLO* e/ou *FUZZY_TC*;

- **Overhead de Roteamento (OR)** - nesta pesquisa o *overhead* de roteamento (Equação 13) foi definido como a média de mensagens de controle emitidas pelos nós *OLSR* durante a simulação [3]. O *overhead* de roteamento é um excelente indicador do desempenho do protocolo de roteamento, isto é, quanto menos mensagens de roteamento forem emitidas mais eficiente o protocolo de roteamento será.

$$OR = \sum_{i=1}^n \text{Mensagens de Controle}. \quad (13)$$

- **Vazão** - é a taxa na qual a rede envia e recebe dados [28]. Por exemplo, a vazão do enlace (veja a Equação 14) usada nesta pesquisa, é definida como a quantidade de bits enviados de um nó

fonte para um nó destino dividida pelo tempo de duração da observação [31];

$$\text{Vazão} = \frac{\text{Dados (Bits) Enviados}}{\text{Tempo Observado}}. \quad (14)$$

7. **Definição de Nós Egoístas:** o tipo de nó egoísta utilizado é aquele que descarta pacotes de dados e retransmite pacotes de controle [18], este tipo de nó egoísta mantém este comportamento todo o tempo. As quantidades de nós egoístas foram selecionadas aleatoriamente no intervalo [0,49]. O comportamento egoísta foi implementado na função *recv* do *OLSR*, Algoritmo 1. Além disso, para contabilizar as perdas de pacotes por egoísmo foi implementado no *NS-2* um evento de descarte denominado *SEL*, que significa *SELFISH*. Por exemplo, se um nó descartar um pacote por egoísmo o *NS-2* gravará no arquivo de *trace* o evento como: **D 19.868681666 _11_ RTR SEL 3445 cbr 1020 [...]** **48:0 53:0 [...]**. Os significados de cada um dos campos do trecho do arquivo de *trace* são apresentados a seguir: **D**: evento de descarte; **19.868681666**: tempo em que o evento ocorreu; **_11_**: nó onde o evento ocorreu; **RTR**: camada em que o evento ocorreu; **SEL**: o tipo de evento, descarte por egoísmo; **3445**: *id* do pacote que foi descartado; **cbr**: tipo de tráfego, *Constant Bit Rate*; **1020**: tamanho do pacote; **[...]**: evento suprimido (supressão nossa); **48:0**: nó que originou o pacote e porta de origem; **13:0**: nó destino e porta de destino.

ALGORITHM 1

Comportamento Egoísta do Nó *OLSR*.

Input: Pacote de Dados ou Controle.

Output: Encaminhamento, Descarte ou Processamento do Pacote.

```

1: if (Pacote de Dados) then
2:   if (Pacote de Dados foi Originado no nó Atual) then
3:     Encaminha(Pacote);
4:   else
5:     Descarta(Pacote);
6:   end if
7: else
8:   Passe o pacote ao OLSR para processamento;
9: end if

```

Os resultados apresentados das métricas de desempenho avaliadas foram calculados da média aritmética de 12 fluxos de tráfego. O *script* utilizado para extração dessas métricas está disponível em <https://bit.ly/2J3g0jw> (ex., *Github*). A Tabela 6 apresenta um resumo dos parâmetros usados na simulação.

Tabela 6. Resumo dos Parâmetros da Simulação.

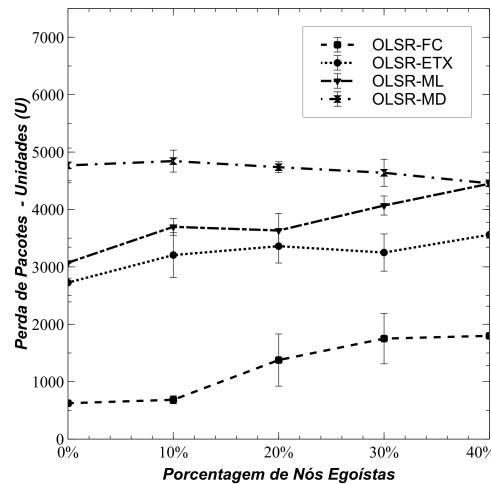
Parâmetros da Simulação	Valor
Área de simulação	$1000m \times 1000m$
Quantidade de nós	50
Tipo de tráfego	CBR UDP
Tamanho dos pacotes	1000 bytes
Taxa de transmissão	16pps (128Kbps)
Quantidades de fluxo de tráfego	12
Modelo de propagação do sinal	TwoRayGround
Carga total de energia do nó	100 Watts (W) (ex., 100J/s)
Potência (TX/RX)	$TX = 1.2W$ e $RX = 0.6W$
Alcance de interferência	250m (Padrão no NS-2)
Tipo de MAC	IEEE 802.11b
Modelo de Mobilidade	Random Waypoint
Velocidade do nó	mínima 5m/s e máxima 15m/s sem pausa
Quantidade de nós egoístas	10%, 20%, 30% e 40%
Comportamento egoísta dos nós	Constante
Tempo de simulação	50s
Willingness OLSR	3
Algoritmo de Seleção de MPR	OLSR-FC = RFC 3626; OLSR-ETX, OLSR-ML e OLSR-MD = Ge et al. [3]
Quantidades de execuções	10
Nível de confiança	95%
Quantidade de Pacotes Gerados	7644

5. Resultados e Discussão

5.1 Avaliação da Taxa de Perda de Pacotes Total e Taxa de Perda de Pacotes por Egoísmo

A taxa de perda de pacotes total¹⁴ foi analisada em termos de *Unidades (U)* e foram gerados 7644 pacotes de dados em cada simulação. Conforme apresentado na Figura 15, o *OLSR-FC* perdeu menos pacotes de dados que as extensões *OLSR-ETX*, *OLSR-ML* e *OLSR-MD*. Em alguns casos, se comparado ao *OLSR-MD*, por exemplo, no cenário com 10% de nós egoístas, o *OLSR-FC* tem um desempenho sete vezes melhor, ou seja, perde 685 (8,96%) pacotes contra 4843 (63,35%) pacotes perdidos pelo *OLSR-MD*, uma diferença de 707%. Entretanto, à medida que aumenta a quantidade de nós egoístas também aumentam as perdas de pacotes do *OLSR-FC* por eventos do tipo *NRTE*, *TTL*, *LOOP* e *IFQ*. Isso pode ser explicado pelo uso da regra 1: *IF IPP is High \Rightarrow Fuzzy Cost is High* (Tabela 1), que faz o *OLSR-FC* ignorar rotas com altas taxas de perdas de pacotes de dados, e em função disso um pacote pode ser redirecionado mais que o necessário e a consequência é o aumento da perda de pacotes pelos eventos supracitados. Todavia, mesmo no cenário com 40% de nós egoístas o *OLSR-FC* perdeu apenas 1799 (23,53%) de pacotes de dados, enquanto as extensões avaliadas perderam: *ETX* = 3560 (46,57%), *ML* = 4449 (58,20%) e *MD* = 4456 (58,29%) unidades de pacotes de dados.

¹⁴Para contabilizar a perda de pacotes total foi levado em consideração o descarte por egoísmo (ex., implementado no *NS-2* como evento *SEL*) e os seguintes eventos de descartes de pacotes do *NS-2*: *NRTE* (não há rota disponível), *LOOP* (loop na rota), *TTL* (número máximo de saltos alcançados) *TOUT* (tempo no *buffer* da fila de rede expirado), *CBK* (camada de redes tenta enviar o pacote para a subcamada MAC e o manipulador detecta que ela está ocupada), *IFQ* (não há espaço no *buffer*) e *ARP* (não há espaço na fila do *buffer ARP*).

**Figura 15.** Média da Taxa de Perda de Pacotes Total.

Observa-se na Figura 15 que o *OLSR-MD* tem uma redução na perda total de pacotes a partir de 10% de nós egoístas. A explicação para isso consiste no fato de que à medida que aumentam os nós egoístas na rede mais pacotes de dados são descartados e, consequentemente, há uma redução do congestionamento. Em função disso, a perda de pacotes por eventos tais como *NRTE*, *TTL*, *LOOP* e *IFQ* diminui. Por exemplo, no ambiente sem nós egoístas o *OLSR-MD* perdeu 4768 pacotes de um total de 7644. Entretanto, no ambiente com 40% de nós egoístas o *OLSR-MD* perdeu 4456 pacotes dos quais 2981 foram perdidos por egoísmo e 1475 foram perdidos por outros eventos, isto é, mesmo perdendo muitos pacotes por egoísmo a redução da perda por outros eventos, causados em função do congestionamento, diminuiu o número total de pacotes perdidos. Além disso, o baixo desempenho do *OLSR-MD* pode ser explicado pelo excesso de mensagens de controle que ele envia para calcular o atraso do enlace, mais que o dobro das outras extensões (Figura 20). Esse excesso sobrecarrega o tráfego causando perdas por congestionamento. Já o baixo desempenho da métrica *ML* pode ser explicado pela instabilidade no cálculo da mesma, que pode ocultar a baixa qualidade de alguns enlaces que fazem parte da rota analisada [16]. Por exemplo, tanto a métrica *ML* quanto a *ETX* são calculadas por meio do envio de mensagens de *probe* (ex., *HELLO*), todavia, os nós egoístas usados nesta pesquisa não descartam mensagens de controle, portanto a métrica *ML* pode selecionar rotas com baixa probabilidade de perdas de mensagens de controle mas com alta taxa de perdas de pacotes de dados. No caso da métrica *ETX* o baixo desempenho pode ter sido causado também pela utilização do mínimo número de saltos como critério de desempate, o que, segundo Passos et al. [15] o uso deste recurso, na métrica *ETX*, pode levar a caminhos com altas taxas de perdas de pacotes.

Com relação às perdas ocorridas apenas por egoísmo (Figura 16), o *OLSR-FC* obteve melhor desempenho em todos os cenários, por exemplo, no cenário com 40% de nós egoístas, ele perdeu 157 (2,05%) pacotes de dados enquanto que as outras extensões perderam: *ETX* = 2599 (34,00%), *ML* =

3213 (42,03%) e $MD = 2981$ (38,99%) unidades de pacotes de dados. Embora evite nós egoístas o *OLSR-FC* ainda perde alguns pacotes de dados em função deles, isso acontece porque as mensagens de controle, que sinalizam o *Custo Fuzzy*, são enviadas em intervalos regulares, por exemplo, uma mensagem *FUZZY_TC* é enviada a cada 5 segundos e durante essa janela de tempo um nó pode escolher rotas compostas por nós egoístas. Nesta pesquisa foi utilizada uma taxa de transmissão de 16pps, assim, se pacotes de dados forem enviados durante 4 segundos através de rotas composta por nós egoístas a quantidade de pacotes perdidos será: $16 \times 4 = 64$.

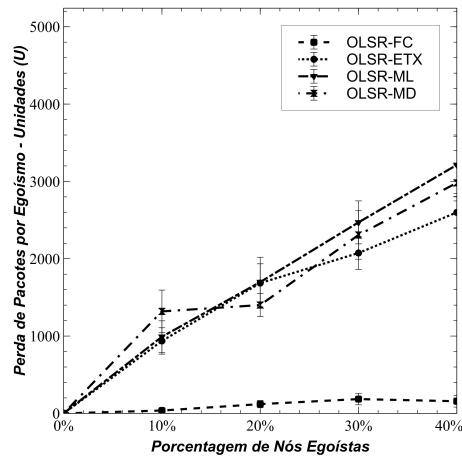


Figura 16. Média da Taxa de Perda de Pacotes por Egoísmo.

5.2 Avaliação do Atraso Fim-a-Fim e Jitter

No que se refere ao atraso fim-a-fim, o *OLSR-FC* perde apenas para a extensão *OLSR-ETX* (Figura 17), no cenário com 40% de nós egoístas. Isso acontece no *OLSR-FC* em decorrência do aumento da carga de processamento, isto é, o pacote fica mais tempo em trânsito até que uma rota livre de nós egoístas esteja disponível, o que provoca o aumento dos atrasos de propagação e processamento. Além disso, a métrica de roteamento *ETX* não encaminha pacotes através de enlaces congestionados e, dessa forma, não deve sofrer oscilações que, às vezes, prejudicam o atraso fim-a-fim [14]. Observa-se também que o *OLSR-FC* em todos os cenários não possui atrasos maiores que 0,150s, enquanto que as extensões testadas possuem diversas ocorrências, por exemplo, o *OLSR-ETX* possui três ocorrências com atrasos maiores que 0,150s. A justificativa para isso consiste no uso do parâmetro *Índice de Conectividade (IC)*. Esse parâmetro é o principal responsável pela escolha de rotas com alta conectividade, ou seja, rotas cujos nós possuem o maior número de vizinhos a um salto de distância, e a conectividade é a principal responsável pelo desempenho das *MANETs*. Dessa forma, quanto maior a conectividade maior o número de vizinhos a um salto, o que possibilita mais *enlaces* para escolha da rota, pois quando há múltiplas rotas com o mesmo custo, aquela com menor número de saltos será escolhida; isso diminui o atraso de processamento do enlace e também pode diminuir o atraso de propagação.

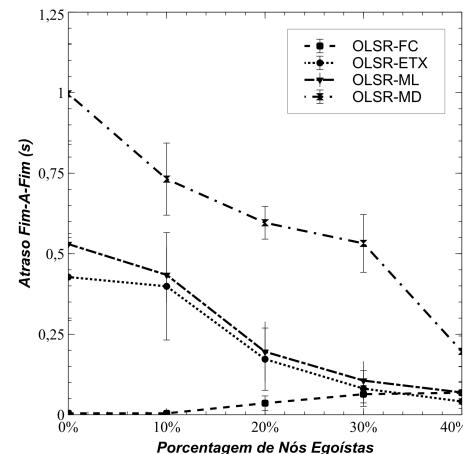


Figura 17. Média do Atraso Fim-a-Fim.

No caso do *Jitter*, variação do atraso (Figura 18), o *OLSR-FC* obteve um *Jitter* baixo e com pouca variação em todos os cenários e esse desempenho favorável pode ser justificado também pelo uso do parâmetro *Índice de Conectividade (IC)*. Todavia, como no atraso fim-a-fim, o *OLSR-FC* perde para o *OLSR-ETX* no cenário com 40% de nós egoístas, como já mencionado na explicação do atraso. Outra observação feita, consiste no fato de que o *atraso médio da rede* diminui à medida que a porcentagem de nós egoístas aumenta. A explicação para isso consiste na redução do número de pacotes de dados em trânsito, que foram descartados pelos nós egoístas. Dessa forma, o tráfego de pacotes diminui provocando a redução do atraso fim-a-fim nas extensões *OLSR-ETX*, *OLSR-ML* e *OLSR-MD*. Outra consequência do baixo desempenho, inerente à métrica *MD*, consiste no uso do ajuste de atraso chamado *smooth factor*, um fator de ajuste de retardo de transmissão que pode ser configurado com valores do seguinte intervalo $]0,1[$. Neste trabalho foi usado o valor 0,4, pois foi o valor que a métrica *MD* obteve seu melhor desempenho em [16]. Nas simulações realizadas, não foram feitos testes para verificar qual valor de *smooth factor* seria ideal ao *OLSR-MD*. Devido a isso, o baixo desempenho do mesmo pode ter sido causado pelo uso do *smooth factor* inadequado.

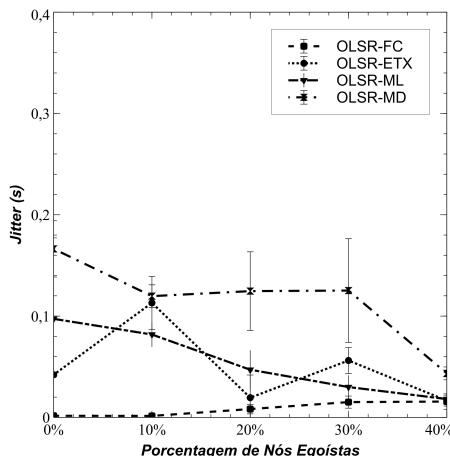


Figura 18. Média do Jitter.

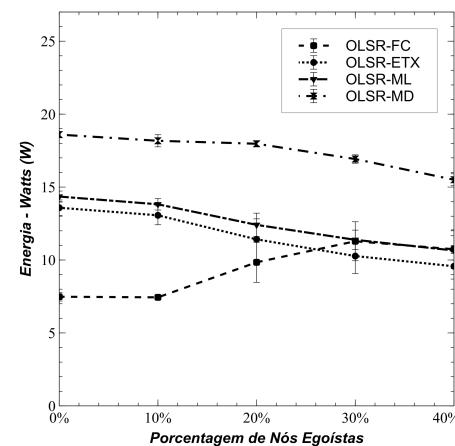


Figura 19. Média do Consumo de Energia.

5.3 Avaliação do Consumo de Energia e Overhead de Roteamento

O consumo médio de energia foi analisado em termos de Watts (W). A Figura 19 mostra que o *OLSR-FC*, nos cenários com até 20% de nós egoístas, consome menos energia que as extensões testadas, por exemplo, o *OLSR-FC* consome $9,841W$ enquanto que as métricas avaliadas consomem: $ETX = 11,415W$, $ML = 12,408W$ e $MD = 17,969W$. Isso é devido ao uso do parâmetro *Energia Residual (ER)* que é utilizado por sua heurística de roteamento, que tem como critério escolher rotas com maior capacidade energética. Todavia, ao passo que a porcentagem de nós egoístas aumenta na rede sua eficiência energética diminui. A explicação para isso reside no fato do *OLSR-FC* ter como um de seus objetivos evitar rotas com altas taxas de perda de pacotes de dados, devido a isso um pacote pode ser redirecionado mais que o necessário, principalmente nos cenários com mais de 20% de nós egoístas, e a consequência disso é o aumento do consumo de energia. O *OLSR-FC* é mais eficiente energeticamente que as outras extensões, embora consuma mais energia que o *OLSR-ETX* e o *OLSR-ML* no cenário com 40% de nós egoístas. Por exemplo, ele gasta $1,167W$ a mais que o *OLSR-ETX* e $0,085W$ a mais que o *OLSR-ML*. Entretanto, no cenário sem nós egoístas, o *OLSR-FC* consome $6,101W$ a menos que o *OLSR-ETX* e $6,868W$ a menos que o *OLSR-ML*.

Conforme apresentado na Figura 19, o *OLSR-MD* tem o maior consumo de energia e isso ocorre devido à quantidade excessiva de mensagens de controle enviadas (*overhead*), o dobro das outras extensões (Figura 20). Além disso, pode ser observado na Figura 19 que o consumo médio de energia das extensões testadas, *OLSR-ETX*, *OLSR-ML* e *OLSR-MD*, diminui. Isso acontece ao passo que a proporção de nós egoístas aumenta na rede, e em função disso, mais pacotes de dados são descartados e, consequentemente, ocorrem menos redirecionamentos desses pacotes. Dessa forma, há uma diminuição do tráfego e tanto os nós egoístas quanto os nós cooperativos economizam energia.

O *overhead* de roteamento foi analisado em termos de unidades, e conforme apresentado na Figura 20, o *OLSR-FC*, no cenário sem nós egoístas, gera 341 pacotes de controle a menos que o *OLSR-ETX*, 354 menos que *OLSR-ML* e 5419 menos que *OLSR-MD*. O *OLSR-FC* gera mais pacotes de controle que o *OLSR-ETX* e o *OLSR-ML* em apenas dois casos: no cenário com 30% de nós egoístas em que ele gera 128 mensagens de controle a mais que o *OLSR-ETX* e 27 a mais que o *OLSR-ML*; e no cenário com 40% de nós egoístas o *OLSR-FC* gera 31 pacotes de controle a mais que o *OLSR-ETX*.

As mensagens de controle geradas a mais pelo *OLSR-FC* são causadas devido ao aumento dos redirecionamentos de pacotes de dados, principalmente nos cenários com 30% e 40% de nós egoístas e as consequências são: congestionamento, colisões e enlaces falhos (ex., perda de enlaces simétricos e/ou enlaces quebrados). Devido a isso, é necessário recalcular os *MPRs* [32] e mais mensagens de controle são geradas [3].

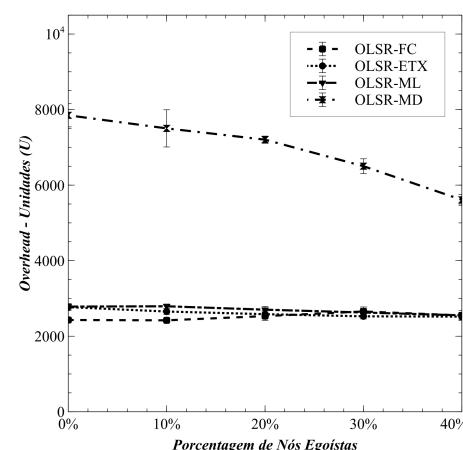


Figura 20. Média do Overhead de Roteamento.

O pior desempenho de todas as métricas testadas foi o do *OLSR-MD*. Isso pode ser explicado pelo fato da métrica *MD* enviar pares de pacotes de controle adicionais para medir o atraso do enlace [16]. Foi observado também que o *OLSR-*

MD tende a diminuir a quantidade de mensagens de controle emitidas à medida que a porcentagem de nós egoístas aumenta na rede. A explicação para este fato está na métrica *MD*, pois essa métrica envia pares de pacotes para calcular o atraso do *enlace* [16]. Sem nós egoístas na rede o tráfego de pacotes de dados em trânsito junto com o *overhead* excessivo causam colisões e muitos pacotes de controle (utilizados para realizar a medição do atraso) são perdidos. Devido a isso, mais pacotes de controle são enviados. Com o aumento dos nós egoístas, o número de pacotes de dados em trânsito diminui, ocorrem menos colisões, e menos pacotes de sondagem são enviados para medir o atraso. Isso explica porque o *OLSR-MD* tende a diminuir o número de mensagens de controle, nos cenários com maior porcentagem de nós egoístas.

5.4 Avaliação da Vazão

Conforme mostrado na Figura 21, o *OLSR-FC* obteve um melhor desempenho nesta métrica, em todos os cenários, com e sem nós egoístas. Por exemplo, no cenário sem nós egoístas, o *OLSR-FC* obteve 123,567 kbps enquanto que as outras extensões obtiveram: *OLSR-ETX* 89,043 kbps, *OLSR-ML* 82,584 kbps e *OLSR-MD* 57,600 kbps. No cenário com 40% de nós egoístas o *OLSR-FC* obteve 114,591 kbps e as demais extensões obtiveram: *OLSR-ETX* 82,450 kbps, *OLSR-ML* 73,541 kbps e *OLSR-MD* 78,382 kbps. O bom desempenho da métrica *FC* se deve ao fato do uso do *Índice de Conectividade (IC)* dos nós como um dos parâmetros de roteamento. Esse índice permite escolher caminhos compostos por nós encaminhadores que possuem maior quantidade de vizinhos a um salto de distância, isto é, maior conectividade, pois, segundo Carey [33], a conectividade da rede é um pré-requisito para fornecer *QoS* porque sem conectividade a vazão pode chegar a zero, no mínimo entre um par de nós.

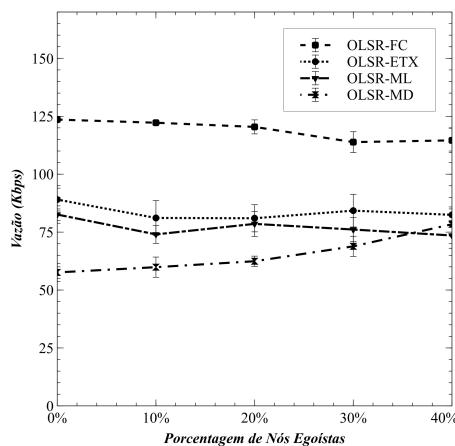


Figura 21. Média da Métrica de Desempenho Vazão.

Observa-se na Figura 21 que, conforme aumentam os nós egoístas, o *OLSR-MD* aumenta sua vazão. Isso pode ser explicado pela redução do tráfego de pacotes de dados que são descartados pelos nós egoístas. Assim, o congestionamento é reduzido, menos colisões ocorrem e consequentemente a

vazão aumenta. Além disso, as extensões testadas têm suas heurísticas de roteamento baseadas no envio de mensagens de *probe*, e isso não acontece no *OLSR-FC* que calcula o *Custo Fuzzy* com base em informações locais. Dessa forma, em cenários com alta mobilidade como este, calcular o *ETX*, o *ML* e o *MD* pode se tornar uma tarefa difícil em função da perda dos pacotes de *probe*.

6. Conclusão e Trabalhos Futuros

Este artigo apresentou uma proposta de extensão ao protocolo *OLSR* denominada *OLSR-FC* que faz uso de um *Sistema de Inferência Fuzzy (SIF)* e utiliza múltiplos parâmetros como o índice de perda de pacotes, a energia residual e índice de conectividade dos nós para encontrar rotas com poucas perdas de pacotes, alta capacidade energética e melhor conectividade. A extensão proposta apresentou melhores resultados que as extensões avaliadas (*OLSR-ETX*, *OLSR-ML* e *OLSR-MD*) com relação às seguintes métricas de desempenho: taxa de perda de pacotes, atraso fim-a-fim, *Jitter*, consumo de energia, *overhead* de roteamento e vazão. Por meio de simulações usando o *NS-2*, verificou-se que o *OLSR-FC* conseguiu evitar rotas compostas por nós egoístas, a principal contribuição. Constatou-se, também, que o *OLSR-FC* é capaz de melhorar a qualidade do tráfego, mesmo na presença de nós egoístas. Por exemplo, no pior cenário, 40% de nós egoístas, ele perde apenas 23,53% dos pacotes enquanto que a extensão melhor colocada, *OLSR-ETX*, perde 46,57%. As únicas situações nas quais o *OLSR-FC* sai em desvantagem, em função do aumento dos redirecionamentos, são: no atraso, perdendo para o *OLSR-ETX* apenas no cenário com 40% de nós egoístas; e nas métricas de desempenho consumo de energia e *overhead* de roteamento em que ele perde nos cenários com mais de 20% de nós egoístas.

Como trabalhos futuros pretende-se adaptar a métrica *FC* para ser usada na seleção dos nós *MPRs* e ajustar o sistema de inferência *Fuzzy* para ser usado em cenários com qualquer quantidade de nós.

Agradecimentos

Os autores gostariam de agradecer à *Fundação de Amparo à Pesquisa do Estado de Mato Grosso (FAPEMAT)* pelo suporte financeiro a esta pesquisa e ao pesquisador Anderson Gusmão pela colaboração na implementação do *SIF* no *MATLAB*.

Contribuições dos Autores

Os autores contribuíram igualmente na realização do trabalho.

Referências

- [1] SINGH SUDHA, D. K. S.; MONDAL, M. S. *Potential Area of Research in MANET In Technological Advancements and Applications in Mobile Ad-Hoc Networks: Research Trends*. 1. ed. India: Kamaljit I. Lakharia, 2012. v. 1.

- [2] BOUKERCHE, A. et al. Routing protocols in ad hoc networks: A survey. *Comput. Netw.*, v. 55, n. 13, p. 3032 – 3080, 2011.
- [3] GE, Y.; KUNZ, T.; LAMONT, L. Quality of service routing in ad-hoc networks using olsr. In: . Big Island, HI, USA: IEEE, 2003. (HICSS, v. 36), p. 9 pp.–.
- [4] KOTHARI, R.; DEMBLA, D. Implementation of black hole security attack using malicious node for enhanced - dsr routing protocol of manet. *Int. J. Comput. Appl. N. Y. USA*, v. 64, n. 18, p. 1–8, 2 2013.
- [5] ZAYANI, M.; ZEGHLACHE, D. Cooperation enforcement for packet forwarding optimization in multi-hop ad-hoc networks. In: . Shanghai, China: IEEE, 2012. (WCNC, '12), p. 1915–1920.
- [6] BUTTYAN, L.; HUBAUX, J.-P. *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. 1. ed. New York, NY, USA: Cambridge University Press, 2007. v. 1.
- [7] WANKHADE, S. V. 1. *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)*, v. 1, n. 1, p. 1–5, 7 2012.
- [8] ROBERT, J.-M.; OTROK, H.; CHRIQUI, A. Rbc-olsr: Reputation-based clustering olsr protocol for wireless ad hoc networks. *Comput. Commun.*, v. 35, n. 4, p. 487 – 499, 2012.
- [9] TOH, C. K. et al. The controversy of selfish nodes in ad hoc networks. In: . Phoenix Park, South Korea: IEEE, 2010. (ICACT '10, v. 2), p. 1087–1092.
- [10] CHARILAS, D. E.; GEORGILAKIS, K. D.; PANAGOPOULOS, A. D. Icarus: hybrid incentive mechanism for cooperation stimulation in ad hoc networks. *Ad Hoc Netw.*, v. 10, n. 6, p. 976 – 989, 2012.
- [11] ZADEH, L. A. Fuzzy sets*. *Inf. Control*, v. 8, n. 3, p. 338–353, 1965.
- [12] CLAUSEN, T. H.; JACQUET, P. *RFC 3626 OLSR - Optimized Link State Routing Protocol*. Reston, USA: The Internet Society, 2003.
- [13] FOUNDATION, N. S. *VINT Project. The Network Simulator - ns-2*. 2009. Disponível em: <<https://www.isi.edu/nsnam/ns/>>.
- [14] COUTO, D. S. J. D. et al. A high-throughput path metric for multi-hop wireless routing. In: . San Diego, CA, USA: ACM, 2003. (MobiCom, '03), p. 134–146.
- [15] PASSOS, D. et al. Mesh network performance measurements. In: . Cuiabá, MT - Brazil: I2TS, 2006. (I2TS, v. 5th), p. 1–8.
- [16] CORDEIRO, W. et al. Providing quality of service for mesh networks using link delay measurements. In: . Honolulu, HI, USA: IEEE, 2007. (ICCCN, '07), p. 991–996.
- [17] BABAKHOUYA, A.; CHALLAL, Y.; BOUABDAL-LAH, A. A simulation analysis of routing misbehaviour in mobile ad hoc networks. In: . Cardiff, UK: IEEE, 2008. v. 1, p. 592–597.
- [18] YOKOYAMA, S. et al. Evaluation of the impact of selfish nodes in ad hoc networks and detection and countermeasure methods. In: . Nara, Japan: IEEE, 2006. (MDM, '06), p. 95–95.
- [19] GOMES, R. L. et al. Using fuzzy link cost and dynamic choice of link quality metrics to achieve qos and qoe in wireless mesh networks. *J. Netw. Comput. Appl.*, v. 34, n. 2, p. 506 – 516, 2011.
- [20] SANTHI, G.; NACHIAPPAN, A. Fuzzy-cost based multiconstrained qos routing with mobility prediction in manets. *Egypt. Inform. J. - Elsevier*, v. 13, n. 1, p. 19–25, 1 2012.
- [21] LETICHEVSKY, A. C.; VELLASCO, M. M. B. R.; TANDESCHEIT, R. Um sistema fuzzy de suporte à decisão para meta-avaliação uma nova abordagem e um estudo de caso desenvolvidos no brasil. *Am. Eval. Assoc. Conf.: Conseq. Eval. Portland Or. U. S.*, v. 15, n. 56, p. 447–462, 7 2006.
- [22] ROSS, T. J. *Fuzzy Logic With Engineering Applications*. 3th. ed. USA: Jhon Wiley & Sons ltd., 2010. v. 1.
- [23] ZHAO, J.; BOSE, B. K. Evaluation of membership functions for fuzzy logic controlled induction motor drive. In: . Sevilla, Spain: IEEE, 2002. (IECON, vol.1), p. 229–234.
- [24] REZENDE, S. O. *Sistemas Inteligentes: Fundamentos e Aplicações*. 1. ed. Barueri, SP: Editora Manole Ltda, 2003. v. 1.
- [25] SNIEDOVIC, M. Dijkstra's algorithm revisited: the dynamic programming connexion. *J. Control Cybern.*, v. 35, n. 3, p. 599 – 620, 2006.
- [26] LEÓN, P. et al. Fuzzy controller developed in a plc, based on weighted average method. In: . Santiago, Chile: IEEE, 2015. (CHILECON, '015), p. 55–60.
- [27] HYYTIA, E. *Random Waypoint Model*. 1. ed. Helsinki, Finland: Helsinki University of Technology, 2005. v. 1.
- [28] SALLEH, A. U. et al. Trace analyzer for ns-2. In: . Selangor, Malaysia: IEEE, 2006. (SCOReD, '06), p. 29–32.
- [29] CHADDA, A. *Quality of Service Testing Methodology*. Tese (Doutorado), New Hampshire, USA, 2004.
- [30] CUNHA, D. de O.; COSTA, L. H. M. K.; DUARTE, O. C. M. B. Uma análise do consumo de energia em redes ad-hoc. In: . Santa Rita do Sapucaí - MG - Brasil: INATEL, 2004. v. 7, p. 39–47.
- [31] ISSARAIYAKUL, T.; HOSSAIN, E. *Network Simulator 2 Ultimate: Post processing throughput calculation*. 2. ed. New york, USA: Springer, 2009. v. 1.
- [32] JACQUET, P. et al. *Optimized Link State Routing Protocol*. Reston, USA: The Internet Society, 2001.
- [33] CAREY, M. F. *Connectivity, throughput, and end-to-end latency in infrastructureless wireless networks with beamforming-enabled devices*. Tese (Doutorado), USA, 2011.