

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Rodrigo Gomes de Jesus

**Aplicação do protocolo RPL em cenários de mobilidade
da Internet das Coisas**

São Carlos

2019

Rodrigo Gomes de Jesus

**Aplicação do protocolo RPL em cenários de mobilidade
da Internet das Coisas**

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Rogério Andrade Flauzino
Coorientadora: Profa. Tania Regina Tronco

**São Carlos
2019**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

G631a Gomes de Jesus, Rodrigo
 Aplicação do protocolo RPL em cenários de
mobilidade da Internet das coisas / Rodrigo Gomes de
Jesus; orientador Rogerio Andrade Flauzino;
coorientadora Tania Regina Tronco. São Carlos, 2019.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2019.

1. RPL. 2. Internet das coisas. 3. VANETs. 4.
Mobilidade. 5. Sistema Fuzzy. 6. Cooja. 7. Contiki. I.
Título.

FOLHA DE APROVAÇÃO

Nome: Rodrigo Gomes de Jesus

Título: "Aplicação do protocolo RPL em cenários de mobilidade da Internet das Coisas"

Trabalho de Conclusão de Curso defendido e aprovado
em 14 / 6 / 2019,

com NOTA 10,0 (Dez, zero), pela Comissão Julgadora:

Prof. Associado Rogério Andrade Flauzino - Orientador - SEL/EESC/USP

Dra. Tania Regina Tronco - SEL/EESC/USP

Profa. Dra. Mônica de Lacerda Rocha - SEL/EESC/USP (docente aposentada)

*Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino*

Primeiramente agradeço a minha mãe, Roseli, que sempre disponibilizou o que podia e não podia nos meus estudos. Com todo seu amor, carinho e muito suor, proporcionou uma mudança na minha vida e de toda a nossa família. Sou grato ao seu apoio desde o início da minha educação, me ensinando desde cedo a acreditar no meu potencial e encarar a vida da forma mais positiva possível. Uma mulher admirável e guerreira que conseguiu atingir o objetivo de ver seus dois filhos formados, tendo boa parte do mérito dessas conquistas.

Agradeço a minha irmã, Bruna, que sempre compartilhou comigo momentos inesquecíveis e que será uma conselheira eterna na minha vida. Seu carinho me fez querer continuar e atingir cada novo objetivo.

Agradeço a todos meus familiares que estiveram presentes nessa fase, dividindo a saudade e a felicidade de curtos finais de semanas que tentei ao máximo estar presente. Estar longe de vocês é um desafio diário na minha vida. Muito obrigado por todo amor.

Agradeço à Marina, quem divide comigo 9 anos de companheirismo, evolução, compreensão e amor. Sua presença me fortaleceu quando mais precisei. Sem ela boa parte das minhas conquistas não teriam ocorrido.

Agradeço aos meus amigos da 012 e de São Carlos, responsáveis por me fazerem desenvolver o meu melhor e dividirem esses anos inesquecíveis. Cada momento com vocês foi único e espero tê-los comigo por muitos anos.

Dedico esse e futuros projetos a todos que positivamente passaram por mim e que mudaram cada vez mais a minha forma de encarar a vida.

AGRADECIMENTOS

Agradeço aos meus familiares que estiveram sempre me apoiando em situações de desafio e crescimento. O amor e empatia de todos vocês foi essencial em todo andamento do projeto.

Agradeço imensamente meus amigos, quem compartilho todas vitórias e momentos de dificuldades acadêmicas, profissionais e pessoais.

Agradeço à professora Tania por me auxiliar na definição do tema e por todo apoio no projeto desde o início. Seus conselhos e *feedbacks* foram muito importantes e espero que eu tenha respondido à altura com esse projeto.

Agradeço ao professor Rogério Flauzino pela orientação e todo corpo docente do departamento de Engenharia Elétrica.

“Quem não se movimenta não sente as correntes que o prendem”

Rosa Magalhães

RESUMO

de JESUS, R. G. **Aplicação do protocolo RPL em cenários de mobilidade da Internet das Coisas**. 2019. 106p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

Este trabalho tem como objetivo a aplicação do protocolo de roteamento de Internet das Coisas para redes de baixa potência com perdas (RPL) em redes de sensores sem fio, utilizando cenários de mobilidade. Dentre os desafios dessa implementação estão a falta de padronização e limitações desse protocolo em algumas de suas topologias. Com base em estudos de sistemas de inferência fuzzy e com gerador de mobilidade *Manhattan grid* pôde-se avaliar métricas como atraso de ponta a ponta, ETX (Número esperado de transmissões) e consumo de energia, comparando as diferentes funções objetivo do protocolo. Variando os parâmetros nas regras de inferência do sistema fuzzy e as diferentes frequências de envio de pacotes, foi verificado que a mobilidade afetou negativamente o desempenho do protocolo, mas o comportamento melhora quando utilizado o método fuzzy. O impacto da velocidade também é avaliado sob o aspecto do consumo de energia dos nós da rede.

Palavras-chave: RPL, Internet das Coisas, VANETs, Mobilidade, Sistema Fuzzy, COOJA, Contiki.

ABSTRACT

de JESUS, R. G. **Study of the RPL protocol in mobility scenarios of Internet of Things**. 2019. 106p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

This work has as a goal the application the Internet routing protocol of Things for Low Power Lossless Networks (RPL) in wireless sensor networks, using mobility scenarios. Among the challenges of this implementation are the lack of standardization and limitations of this protocol in some of its topologies. Based on studies of fuzzy inference systems and with Manhattan mobility, it was possible to evaluate metrics such as end-to-end delay, ETX (Expected transmission Count) and energy consumption, comparing the different objective functions of the protocol. By varying the parameters in the rules of inference of the fuzzy system and the different frequencies of sending of packages, it was verified that the mobility affected negatively the performance of the protocol, but the behavior improves when using the fuzzy method. The impact of speed is also evaluated in terms of the power consumption of the network nodes.

Keywords: RPL, Internet of Things, VANETs, Mobility, Fuzzy System, COOJA, Contiki.

LISTA DE FIGURAS

Figura 1 – Crescimento da Internet em 2019	30
Figura 2 – Estatística geral da internet em 2019	30
Figura 3 – Três paradigmas da Internet das coisas	33
Figura 4 – Crescimento da IoT - Número de dispositivos conectados	35
Figura 5 – As barreiras mais significantes na adoção da IoT nos processos internos	36
Figura 6 – Investimento em IoT nos mais diversos setores industriais	37
Figura 7 – Protocolos de rede IoT mapeados no modelo TCP/IP	41
Figura 8 – Camadas e Arquitetura IoT	44
Figura 9 – Instância de uma RPL	45
Figura 10 – Cabeçalho mensagem DIO	46
Figura 11 – Operação após recebimento da DIO	47
Figura 12 – Nós e <i>Delay End-to-end</i>	51
Figura 13 – Função de pertinência	51
Figura 14 – Sistema de Inferência Fuzzy	52
Figura 15 – Sistema Completo de Inferência Fuzzy	53
Figura 16 – As redes ad-hoc veiculares (VANETs)	55
Figura 17 – Descrição do <i>sky mote</i> com <i>sky mote ID</i> , <i>sky mote type</i> , identificador, fonte <i>contiki</i> e <i>firmware</i>	60
Figura 18 – Módulo de Mobilidade Implementado no Simulador Cooja	60
Figura 19 – Ambiente 500 metros quadrados contendo 25 nós	61
Figura 20 – Cenário com os nós enviado/recebendo pacotes e as posições indicadas de cada um dos sensores	62
Figura 22 – Cenário de mobilidade Manhattan	62
Figura 21 – Ambiente de simulação com os nós e a área de abrangência do nó 24 .	63
Figura 23 – Estrutura do processo Fuzzy	65
Figura 24 – Função Pertinência para <i>Delay</i>	66
Figura 25 – Função Pertinência para <i>ETX</i>	68
Figura 26 – Função Pertinência para Número de Saltos	70
Figura 27 – Função Pertinência da Qualidade do nó vizinho - Saída do sistema . . .	74
Figura 29 – Comparativo da taxa de entrega de pacote com 10 nós móveis	78
Figura 28 – Comparativo da taxa de entrega de pacote com 5 nós móveis	78
Figura 30 – Comparativo da taxa de entrega de pacote com 20 nós móveis	79
Figura 31 – Comparativo do <i>delay</i> na entrega do pacote com 5 nós móveis	81
Figura 32 – Comparativo do <i>delay</i> na entrega do pacote com 10 nós móveis	81
Figura 33 – Comparativo do <i>delay</i> na entrega do pacote com 20 nós móveis	82

Figura 34 – Consumo médio de energia nos nós. Os 3 cenários propostos e a influência da velocidade dos nós móveis.	84
Figura 35 – Inicialização da rede	105
Figura 36 – Iniciando uma simulação	105
Figura 37 – Parâmetros e criação da simulação	105
Figura 38 – Processo de Simulação	106
Figura 39 – Simulação da rede e paine principal	106
Figura 40 – Simulação da rede RPL	106

LISTA DE TABELAS

Tabela 1 – Sistemas de Mote e Microcontrolador disponíveis no Simulador Cooja	59
Tabela 2 – Fonte: Adaptado de (INCORPORATED, 2017)(INCORPORATED, 2002)	59
Tabela 3 – Memória dos Microcontroladores disponíveis no Simulador Cooja	59
Tabela 4 – Fonte: Adaptado de (INCORPORATED, 2017)(INCORPORATED, 2002)	59
Tabela 5 – Regras de avaliação utilizadas no processo de Inferência	71
Tabela 6 – Obtendo valor de saída de Qualidade do nó - Regra 1	72
Tabela 7 – Obtendo valor de saída de Qualidade do nó - Regra 2	73
Tabela 8 – Obtendo valor de saída da Qualidade do nó - Regra 3	74
Tabela 9 – Levantamento dos valores de PDR obtidos nos cenários propostos	80
Tabela 10 – Levantamento dos valores de <i>delay</i> obtidos nos cenários propostos	83
Tabela 11 – Levantamento dos valores de consumo médio de energia nos nós obtidos nos cenários propostos	85

LISTA DE ABREVIATURAS E SIGLAS

LPWN	<i>Low-Power Wireless Networks</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
6LoWPAN	<i>IPv6 over Low Power and Lossy Networks</i>
IETF	<i>Internet Engineering Task Force</i>
ROLL	<i>Routing Over Low-power and Lossy networks</i>
RPL	<i>Routing Protocol for Low power and Lossy links</i>
DODAG	<i>Destination oriented Directed Acyclic Graph</i>
VANET	<i>Vehicular Ad-hoc Network</i>
AP	<i>Access Point</i>
MN	<i>Mobile node</i>
MANET	<i>Mobile Ad-Hoc Network</i>
ICMPv6	<i>Internet Control Message Protocol version 6</i>
DIO	<i>DODAG Information Object</i>
DIS	<i>DODAG Information Solicitation</i>
DAO	<i>Destination Advertisement Object</i>
DAO-Ack	<i>Destination Advertisement Object-Acknowledgement</i>
ETX	<i>Expected Transmission Count</i>
HPC	<i>Hop Count</i>
IPv6	<i>Internet Protocol Version6</i>
UDP	<i>User Datagram Protocol</i>
TCP	<i>Transmission Control Protocol</i>
ISM	<i>Industrial Scientific and medical Radio Band</i>
RDC	<i>Radio Duty cycling</i>

API	<i>Application Programming Interface</i>
uIPV6	<i>Micro IPv6</i>
OF0	<i>Objective Function Zero</i>
MRHOF	<i>Minimum Rank Hysteresis Objective Function</i>
WLAN	<i>Wireless Local area network</i>

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Contextualização	25
1.2	Justificativa	26
1.3	Objetivos	26
2	CONCEITOS E BASE TEÓRICA	29
2.1	Internet das coisas	29
2.1.1	Contexto atual da IoT	34
2.1.2	Perspectivas e Desafios em IoT	35
2.2	Protocolos e Camadas em IoT	38
2.2.0.1	TCP/IP	38
2.2.1	Protocolos da camada de Aplicação em IoT	40
2.2.1.1	MQTT	40
2.2.1.2	CoAP	41
2.2.2	Protocolos da camada de rede	42
2.2.2.1	IPv6	42
2.2.2.2	6LoWPAN	43
2.3	Protocolo RPL	43
2.3.1	Mensagens de Controle	45
2.3.2	Criação da DODAG	47
2.3.3	Funções objetivo	48
2.3.4	Métricas de avaliação	49
2.3.4.1	Número de saltos	49
2.3.4.2	Energia consumida	49
2.3.4.3	Contagem de Transmissão Esperada (ETX)	50
2.3.4.4	Perda de pacote	50
2.3.4.5	<i>Delay End-to-End</i>	50
2.4	Lógica e Sistema Fuzzy	51
2.5	Cenários de Mobilidade	54
3	MATERIAIS E MÉTODOS	57
3.1	Sistema Operacional Contiki	57
3.2	Simulador Cooja	58
3.3	Simulador gerador de Mobilidade - Modelo Manhattan	59
3.4	Método proposto de Função Objetivo baseado em lógica fuzzy	62
3.4.1	Sistema Fuzzy	64

3.4.2	Composição de métricas e Fuzzyficação	65
3.4.2.1	Métrica <i>Delay End-to-End</i>	66
3.4.2.2	Métrica ETX	67
3.4.2.3	Número de Saltos	69
3.4.3	Motor de Inferência	70
3.4.4	Defuzzyficação	73
4	RESULTADOS E DISCUSSÃO	77
4.1	Avaliação de Performance	77
4.1.1	Taxa de Entrega de pacotes	77
4.1.2	<i>Delay End-To-End</i>	80
4.1.2.1	Consumo de Energia e a influência da velocidade	83
5	CONCLUSÃO	87
	REFERÊNCIAS	89
	APÊNDICES	95
	ANEXOS	99

1 INTRODUÇÃO

1.1 Contextualização

Com o advento da Internet das Coisas (IoT, do inglês *Internet of Things*) prevê-se a conectividade de bilhões de objetos do nosso cotidiano à rede de Internet. Tal conectividade se dá majoritariamente por meio de redes de baixa potência com perdas (LLNs, do inglês *Low Power and Lossy Networks*) que possuem baixa capacidade energética e computacional.

A conexão de diversos elementos, com uso de diferentes tecnologias, possuindo capacidade de processamento, armazenamento e comunicação variados, em uma única rede, apresenta uma série de dificuldades e desafios para a IoT (CHARARA, 2018).

Boa parte desses elementos - como sensores, por exemplo - têm como uma das principais características a escassa quantidade de recursos, exigindo, portanto, o uso em protocolos e aplicações muito bem otimizados a fim de conseguir um bom funcionamento dada esta limitação (BRITO, 2019).

Nesse sentido, dada a quantidade de elementos sendo conectados, há necessidade de um aumento do número de endereços IP (*Internet Protocol*) nesses cenários. Dessa forma, visando a atender a essa necessidade, houve a criação do IPv6 (*Internet Protocol version 6*) e posteriormente o 6LoWPAN (*IPv6 over Low power Wireless Personal Area Networks*) que trazem grande ajuda, viabilizando o endereçamento destes dispositivos e auxiliando no problema de limitação de recursos. (DEERING; HINDEN; NOKIA, 1998) (SHELBY; BORMANN, 2009).

Seguindo essa linha, desenvolveu-se também o RPL (*IPv6 Routing Protocol for Low Power and Lossy Networks*) que tornou-se rapidamente um dos protocolos de roteamento para IoT, incorporando diversas pilhas de outros protocolos (GADDOUR; KOUBÂA, 2012).

A priori, a padronização do RPL não contempla a mobilidade e implementações são necessárias sempre que estudos como esse forem realizados, desde mudanças na topologia da criação dos nós e, principalmente, até alterações das chamadas funções objetivo (OF, *Objective Function*), que determinam a forma e eficiência como os pacotes de dados serão encaminhados do início até o final do caminho (PISTER et al., 2009).

Considerando o comportamento móvel desses objetos conectados, áreas de pesquisa surgiram como em redes veiculares Ad Hoc (VANET), estimuladas pelo aumento do transporte privado e situações das quais a esfera pública e pessoal podem ser beneficiadas. Por se tratarem de características diferenciadas, como topologia altamente dinâmica, conectividade intermitente, mobilidade limitada dos nós, densidade de rede variável, etc.,

as VANETs possuem grandes desafios na comunicação de dados entre os nós (REHMAN et al., 2014).

1.2 Justificativa

Ainda pouco explorado em diversos ambientes de mobilidade, o uso do protocolo RPL em comportamentos móveis - que vêm ganhando mais destaque nas rotinas de uso de tecnologias IoT - é ainda um desafio. Esse comportamento se mantém quando lidamos com estudo de mobilidade urbana, como é o caso da mobilidade Manhattan. (ALAM; SHER; HUSAIN, 2009) Consiste em movimentos lineares em formas de *grid*, movimentando seus elementos para frente, à esquerda e à direita, como ocorre em vias urbanas.

Diante de cenários dinâmicos, decisões de roteamento realizadas de forma pouco inteligentes podem afetar o número de retransmissões de pacotes, exigindo mais esforço em toda rede sem fio, o que resulta em impactos sobre consumo de energia, tempo de entrega, largura de banda e processamento nos elementos.

Com a finalidade de obter melhor performance e validar pontos já discutidos em outros estudos sobre o tema, serão inseridos mecanismo de controle fuzzy no protocolo padrão RPL. Tal controle visa otimizar as decisões na transmissão de dados e todos resultados são validados no Cooja (DESHPANDE, 2015), um simulador baseado em Contiki (CONTIKI-OS, 2013) usado para simular o modelo e quantificar as melhorias obtidas.

1.3 Objetivos

Com base no exposto nas seções anteriores, o objetivo principal desse trabalho é a implementação e análise das métricas envolvendo o protocolo RPL em ambientes móveis no contexto das VANETs. Além disso, pretende-se analisar, via simulação Cooja, o comportamento da função objetivo que utiliza o sistema de inferência fuzzy quando comparado com as duas funções objetivos hoje padrões desse protocolo: OF0 e MRHOF.

Mediante isso, este trabalho pretende:

1. Caracterizar a Internet das Coisas quando aplicada em redes de baixa potência e com perdas e seu protocolo RPL;
2. Extrair informações sobre o sistema de inferência fuzzy e como realizar a aplicação desse controlador no problema proposto, avaliando a qualidade do nó ponderando os elementos de entrada;
3. Investigar, por meio de simulações, o comportamento do protocolo em ambientes móveis comparando as redes de OFs padrões com as OF's implementadas no projeto;

4. Detalhar a análise com variações nos cenários de mobilidade, frequência nos envios de pacotes e velocidade dos nós móveis, ratificando o comportamento do protocolo na mobilidade proposta.

2 CONCEITOS E BASE TEÓRICA

2.1 Internet das coisas

Existente sob o conceito de um novo paradigma e ponto de vista que vem cada vez mais ganhando espaço na realidade da tecnologia sem fio, a Internet das Coisas é caracterizada pela presença de múltiplas conexões em torno do ser humano, como sensores, atuadores, etiquetas RFID (identificadores por radiofrequência), celulares portáteis, dentre outros que, por meio da interação entre si, são capazes de atingir objetivos comuns (ATZORI; IERA; MORABITO, 2012) sendo autônomos, automáticos e sensíveis ao contexto no qual são imersos (LEMOS, 2013).

A definição de IoT segundo (VERMESAN et al., 2011) nos indica que ela torna possível as pessoas e coisas conectadas a qualquer hora, a qualquer coisa, em qualquer lugar, com qualquer coisa e em qualquer um, de preferência usando qualquer caminho/rede e qualquer serviço (Perera et al. 2013). Tais objetos, ao realizarem interações sem a necessidade de *input* humano, geram um fluxo de dados e essa conexão permite que eles transmitam dados para outros objetos no meio, formando assim uma Internet de coisas (AFONSO; PEREIRA; PEREIRA, 2015).

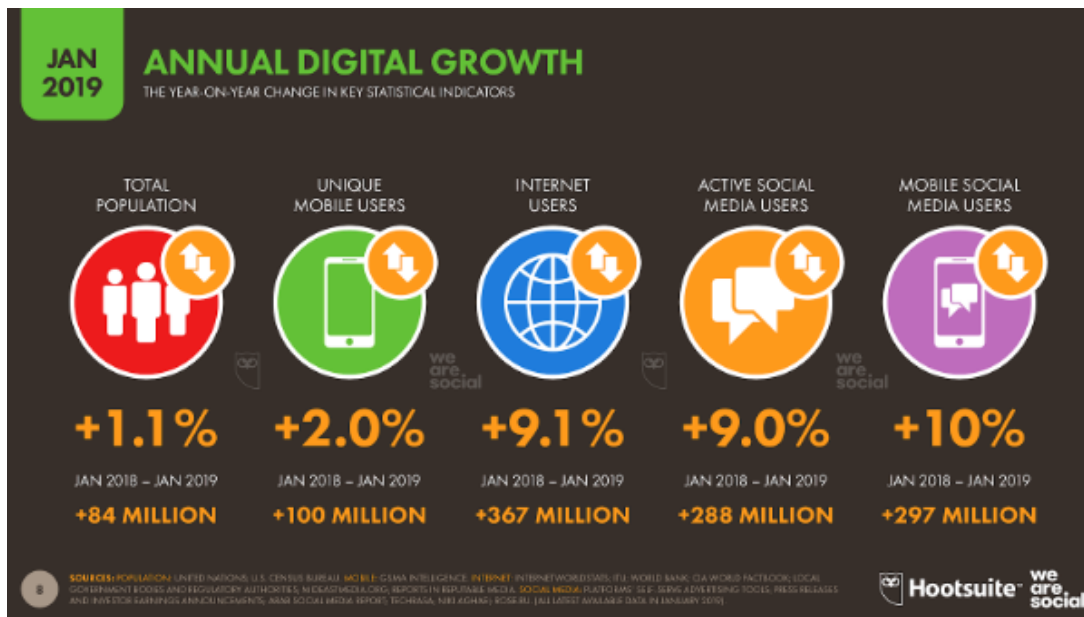
Como aponta o relatório *Digital in* de 2019, mais da metade da população já conta com acesso à internet em todo mundo, exibido no relatório exibido pelos serviços online *Hootsuite e We Are Social*. Como relatado no estudo, mais de 4 bilhões de pessoas conectadas à rede global, diante de estimativas recentes que apresentam uma população mundial de 7,67 bilhões de seres humanos.

Mais precisamente, iniciamos 2019 com 4,38 bilhões de pessoas online (57 % de todas as pessoas do planeta) como visto na Figura 1, um aumento de 9,1% em relação ao ano anterior, como mostrado na Figura 2. As redes sociais são utilizadas por cerca de 3,48 bilhões de pessoas (ou seja, 45 % de todo o mundo), com um crescimento de 9,0% quando comparamos com 2018. (KEMP, 2019).

Nesse sentido, quando expandimos o conceito ao alcance da Internet às coisas, estamos associando à qualquer objeto ou pessoa que esteja capaz de ser distinguido pelo mundo real. Dessa forma, objetos não apenas eletrônicos e presentes na nossa rotina tecnológica, mas inclusive “coisas” que normalmente não estão associadas a esse conceito - podemos citar roupas, móveis, peças e equipamentos, monumentos e obras de arte, produtos, alimentos em geral (KOSMATOS; TSELIKAS; BOUCOUVALAS, 2011)

Em seu início, ainda não denominada como hoje, a IoT restringia-se a mecanismos sem inteligência, porém que tiveram seu potencial explorado em seguida. Conhecida como a Fase 1 da IoT, nesse período havia o conceito criado por Ashton em 1998, cofundador

Figura 1: Crescimento da Internet em 2019



Fonte:(KEMP, 2019)

do Auto-ID Center do Massachusetts Institute of Technology (MIT), quem utilizou pela primeira vez o termo “Internet of Things”, muito vinculado às etiquetas RFID e denominado “Thing-to-Thing Internet” (SANTUCCI, 2009), no qual o mecanismo de radiofrequência das tags era capaz de não só identificar objetos, mas sim tornar tais objetos capazes de

Figura 2: Estatística geral da internet em 2019



Fonte:(KEMP, 2019)

gerir certa inteligência e comunicação. Nesse momento da história, "Internet das Coisas" não era ainda um conceito acadêmico ou científico, tampouco um conceito tecnológico proposto por algum órgão da telecomunicação.

Contemporânea a essa fase está a evolução no estudo das redes de sensores sem fio, as denominadas *Wireless Sensors Network* (WSN), descritas como nós de rede que controlam e detectam cooperativamente o ambiente, com intuito de monitorizar áreas de difícil acesso ou inóspitas, tais como oceanos, desertos, vulcões, florestas, áreas industriais, infraestruturas, entre outros. Com diversas aplicações e oportunidades de negócios surgindo a cada dia tornando possível a interação entre pessoas ou máquinas (computadores, sistemas com capacidade de processamento de dados) e o ambiente no qual estão imersos. (RAWAT et al., 2013) O mercado de WSN tem previsão de atingir 2 bilhões de dólares em 2022, face aos 0,45 bilhão de dólares de 2012 (HARROP; DAS, 2012) (HARROP, 2012).

As WSNs atualmente incluem, para além de sensores, atuadores, gateways (entidade central com interface para Internet, para supervisão/monitorização, para manutenção, entre outros) e cliente final (utilizadores do sistema). Os nós de redes devem ter capacidade de processamento de dados. Para além da aquisição e tratamento dos mesmos, devem ainda possuir capacidade para cooperação com outros nós de rede no sentido de partilha de tarefas e aferição de resultados (dependendo dos algoritmos implementados em cada nós de rede

Nesse primeiro momento, os elementos RFID e a rede de sensores não estavam vinculados mesmo a nível de conceito, estando limitados a elementos estáticos e sem ações ativas, e redes ainda distantes da realidade da população. Segundo a linha do tempo envolvendo o conceito, temos em 2005 uma evolução na ideia de Ashton, consistindo agora em um mundo habitado por um crescente de objetos inteligentes e com comunicação entre si, iniciando um dos desafios envolvendo os tráfego na rede IP existente até então.

Na segunda fase, temos um avanço nos estudos na rede de sensores, adicionando funções de atuadores (*Wireless Sensor Actor Network*, WSAN). Consolidando os comportamentos de rede de sensores, temos aqui a incorporação de ações dentro do campo imerso, com nós atuadores (servomotores, por exemplo) e exigências dentro do ambiente de controle da informação, pois lidamos aqui com comunicações bidirecionais. Espera-se que essa precisão fosse ativada por meio do uso combinado dos sensores no dispositivo. As aplicações dessa capacidade são variadas. Por exemplo, vários indivíduos que usam dispositivos inteligentes podem, com efeito, formar uma WSAN para monitorar populações inteiras.

As WSANs são um elemento da tendência crescente em direção à IoT. Como com muitos outros elementos da IoT, as WSANs levantam questões de privacidade e segurança (ROUSE, 2015)

Sob a ótica operacional, WSANs não podem ser consideradas um aprimoramento ou complemento do sensor sem fio, mas o conceito é um pouco além disso (WANG et al., 2007). Elas são uma nova geração de redes de sensores (T.MELODIA et al., 2007). Enquanto WSANs e WSNs compartilham muitas considerações comuns sobre o design da rede, como confiabilidade, conectividade, escalabilidade e eficiência energética, a coexistência de sensores e atuadores em WSANs causa uma diferença substancial entre esses dois tipos de redes.

À vista disso, WSANs possuem a capacidade de atuar sobre o mundo físico, diferente das WSNs. Nas WSANs, como nas WSNs, há também uma grande preocupação com o consumo de energia visando atender aos requisitos de comunicação confiável e em tempo real (CAYIRCI; COPLU; MIROGLU, 2005).

Em uma perspectiva geral, a topologia básica de uma WSAN é caracterizada por um conjunto de nós conectados por uma rede de radiofrequência (RF), indicada na figura, o qual ilustra três tipos de nós: o nó sensor, também chamado *source*, o qual recebe algum dado físico, transforma-o em um dado digital e, em seguida, o envia para um nó destino, chamado *sink*. O nó *sink*, por sua vez, recebe e processa os dados enviados pelos sensores e envia os mesmos para um nó atuador, este com a capacidade de converter os resultados em ações físicas ou químicas. (SABRI et al., 2011)

Por fim, em sua fase 3, a IoT consolida conceitos como Real World Internet (RWI) e Augmented Reality (RA), ambos a partir de 2009. O primeiro consolida visões de que o mundo é ou pode ser realmente vivenciado por meio da Internet, em tempo real e em qualquer local do mundo. Na RWI assume-se que o fluxo de informações de e para dispositivos IoT está ocorrendo por meio de links de comunicação locais com e sem fio entre dispositivos na proximidade e/ou por meio de interconexões globais na forma de redes móveis e internet atuais ou futuras infraestruturas de rede fixa e móvel

Uma propriedade importante do RWI que se distingue da Internet atual é a heterogeneidade, tanto em relação aos tipos de dispositivos quanto aos protocolos de comunicação utilizados. O IPv6 e, em particular, o 6lowPAN desempenham um papel importante, mas outros protocolos sem fio proprietários também continuarão sendo usados (DOMINGUE et al., 2011). Já a RA realiza o misto entre o real e o virtual, apresentando resultados no qual não temos certeza sobre a sua realidade, tornando-se complexa toda experiência imersiva.

No campo da Arquitetura, pode-se observar a expansão de experiências que exploram como a realidade virtual, aumentada e mista, confundindo as fronteiras entre o físico e o virtual e questionando o que é real e o que é fictício, questionando como podemos interagir com o mundo agora e no futuro próximo. (BLOCK, 2019). No ramo da saúde, temos o crescimento de estudos mostrando que plataformas como videogames utilizando efeitos de realidade virtual e realidade aumentada vêm sendo utilizados como ferramenta de trabalho cotidiana na área, principalmente por fisioterapeutas e terapeutas ocupacionais

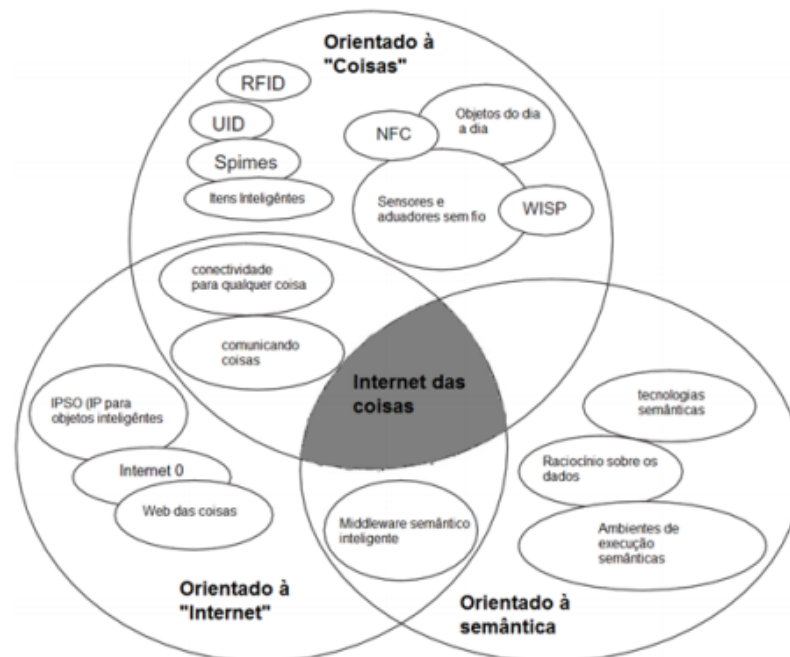
na reabilitação de pacientes. Além desse ambiente, o mesmo pode-se dizer em relação à educação física, que estão fazendo melhorias nos processos e análises no uso das mesmas aplicações (CROCETTA et al., 2015).

Outra área com resultados já consolidados é a educação, apresentando melhor engajamento e interação no aprendizado, somando fatores como o dinamismo, elementos geográficos e rapidez no ensino. Aplicativos como o *ZooKazam* ou *Bugs 3D* são capazes de apresentar modelos 3D de animais, com gráficos e informações sobre as espécies com imagens. (MOBILE, 2018)

Nesse sentido temos a essência da fase 3, marcada pela imersão e complexidade entre os ambientes virtuais e reais.

Como ilustrado na figura 3, a IoT é caracterizada por três principais paradigmas: uma visão focada no termo orientado a “Internet”, outra visão focada no termo orientado a “Coisas” (sensores), onde a pesquisa busca a integração de objetos em uma estrutura comum; e uma terceira visão com foco em questões orientadas a semânticas(conhecimento), do ponto de vista de comunicação entre dispositivos distintos. Visto isso, a IoT pode ser vista como a intersecção desses três paradigmas (ATZORI; IERA; MORABITO, 2012).

Figura 3: Três paradigmas da Internet das coisas



Fonte: Adaptado de (ATZORI; IERA; MORABITO, 2012)

2.1.1 Contexto atual da IoT

A atual fase de evolução na internet das coisas tem a sua essência teórica embasada na Computação Ubíqua, termo este criado por Mark Weiser, um cientista da computação que atuava como CTO (*Chief Technology Officer*) na *Xerox's Palo Alto Research Center*(Parc) e descritos no artigo *The Computer for 21st Century*, publicado em 1991, onde defendia uma visão de futuro em que os computadores seriam invisíveis e incorporados aos objetos do nosso cotidiano (WEISER, 1991).

Substituindo os computadores convencionais (PC's), Weiser afirma que em uma perspectiva futura teremos a era da tecnologia calma, fortalecendo a ideia de que as tecnologias mais profundas são as que desaparecem da visão humana, tecendo-se na malha da vida cotidiana até serem indistinguíveis desta, onde elementos especializados de hardware e software ligados por fio e ondas de rádio, serão tão onipresentes que nem serão notadas a sua existência (WEISER, 1991).

Por conseguinte, ligada ao contexto inaugurado por Weiser, podemos descrever o atual cenário da IoT como uma extensão do sistema de nomes onde é possível associar a identificação digital (por exemplo, endereço IP ou código RFID) e elementos físicos (como um *pallet* em um depósito) de maneira padronizada e há, nesse momento, uma disseminação do que está conectado

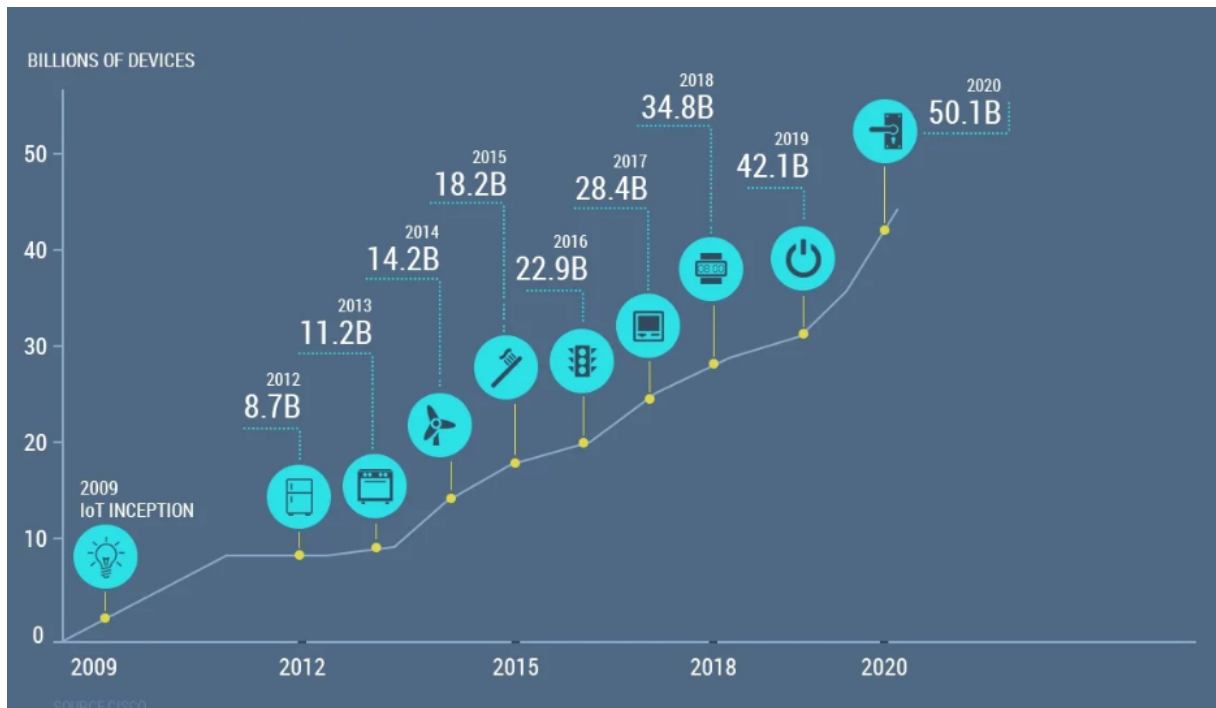
Ainda a própria natureza do ecossistema tecnológico da IoT em sua última fase permanece indefinida. Isso explica porque as fronteiras da IoT e seus mercados estão nebulosos (MADAKAM; SIDDHARTH; TRIPATHI, 2015). Da mesma forma, nenhuma definição única é aceita e compartilhada por sua comunidade de usuários, o que intensifica algumas ausências de limites no seu conceito. Em seu estudo sobre o mercado, Madakam sublinha que muitos grupos diferentes incluindo acadêmicos, profissionais, inovadores, desenvolvedores, o público e organismos profissionais propondo sua própria definição do termo nos dias de hoje, o que se mantém nos últimos anos.

Analisando dados de 2019, a agência americana *Mobidev* afirma, em um relatório recente, que o número de dispositivos conectados em 2019 é de 4,1 bilhões. Com o aumento do número de dispositivos capturando e trocando grandes quantidades de dados, a IoT está abrindo uma infinidade de oportunidades para empresas B2B (*Business to Business*) abrirem as portas de novas fontes de receita.(MITAL, 2018).

Como mostrado na Figura 4 há um crescimento de 7,3 bilhões de dispositivos conectados no último ano, além de uma previsão crescimento de 8 bilhões de objetos para 2020. Quanto ao estudo de mercado, uma pesquisa da IDC (*International Data Corporation*) mostra que 31,4% das empresas aderiram ao caminho dos lançamento de produtos da IoT, enquanto 43% adicionais estavam procurando fazê-lo em 2020. Dos que o utilizam, 55% acreditavam que a IoT representava uma estratégia para ajudá-los a

enfrentar sua concorrência de maneira mais eficiente. (IDC... ,).

Figura 4: Crescimento da IoT - Número de dispositivos conectados



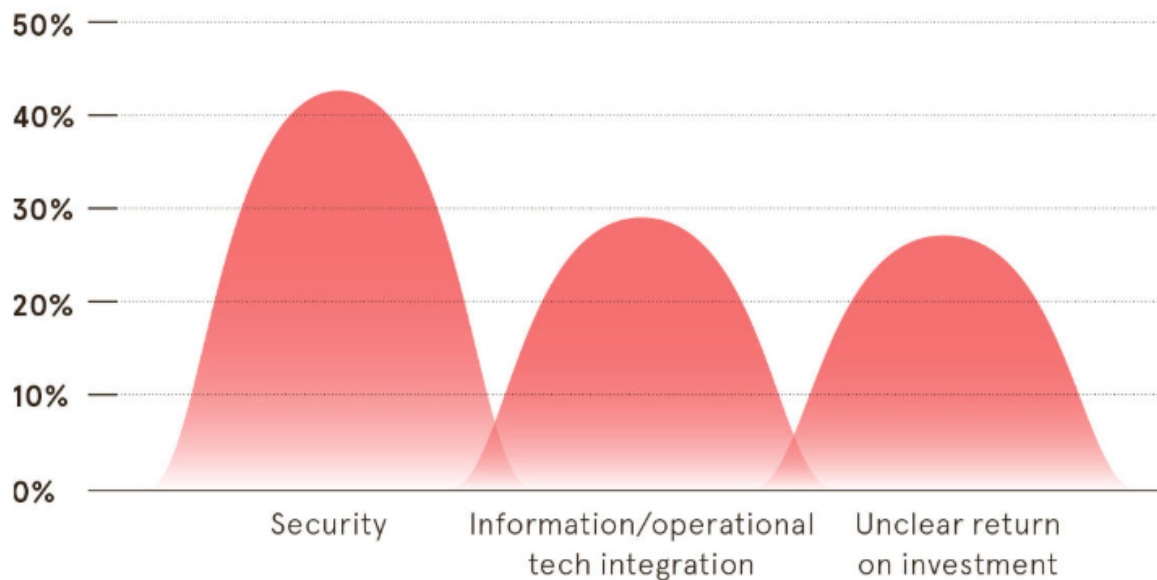
Fonte: (MITAL, 2018)

2.1.2 Perspectivas e Desafios em IoT

Segundo a consultoria *Gartner*, há uma expectativa de um gasto de S\$ 3,76 trilhões em 2019 com Tecnologia da Informação - um aumento de 3,2% em relação a 2018 (BRITO, 2019). Nesse grupo de tecnologias estão aprendizado de máquina (*machine learning*), a inteligência artificial e a Internet das Coisas e, nesse sentido, a perspectiva para a IoT é de crescimento e atenção aos possíveis desafios que surgem com a aplicação desse conceito. Alguns elementos impedem que a IoT tenha maior aderência em aplicações de mercado, principalmente o mercado executivo. Segundo Rob Sheppard, diretor de IoT da *Intel* no Reino Unido, é necessário que os esforços atuais para expansão da tecnologia sejam o de trazer o contexto da Internet das Coisas aos tomadores de decisão executiva em grandes empresas, ou seja, para dentro das salas de reuniões (CHARARA, 2018).

Além desse elemento de insegurança para implantação de tecnologias em IoT, o tempo de implementação também é outro fator que compromete o início da operação. Estima-se que 75% dos projetos levam o dobro do tempo do planejado. Sob a ótica financeira, o custo também é outro fator desafiador. Há um custo por sensor de US\$ 10 a US\$ 50 e, conseqüentemente, ambientes de grande complexidade e projetos de início-fim custam cerca de US\$ 1 milhão (CHARARA, 2018), o que torna a expansão da tecnologia e do conceito limitado a grandes empresas e corporações.

Figura 5: As barreiras mais significantes na adoção da IoT nos processos internos



Fonte: (CHARARA, 2018)

Além disso, a segurança no tráfego de dados entre objetos de IoT - ilustrado na Figura 5 é essencial para o sucesso da operação. Quando o conceito de IoT é aplicado na área de saúde, por exemplo, existem dados realmente importantes que podem ser manipulados por invasores e causarem danos irreversíveis. Estudos e *benchmarks* são, em todos esses pontos, necessários para que exista não só uma aceitação do mercado, mas também condições mais favoráveis a operação, além de profissionais em engenharia de dados e *Customer Success* (sucesso do cliente), capazes de trabalhar com essa infinidade de informações e aplicar aos mais diversos contextos dos usuários e/ou organizações que podem se beneficiar dessa tecnologia. (PICKENS; AIELLO, 2018).

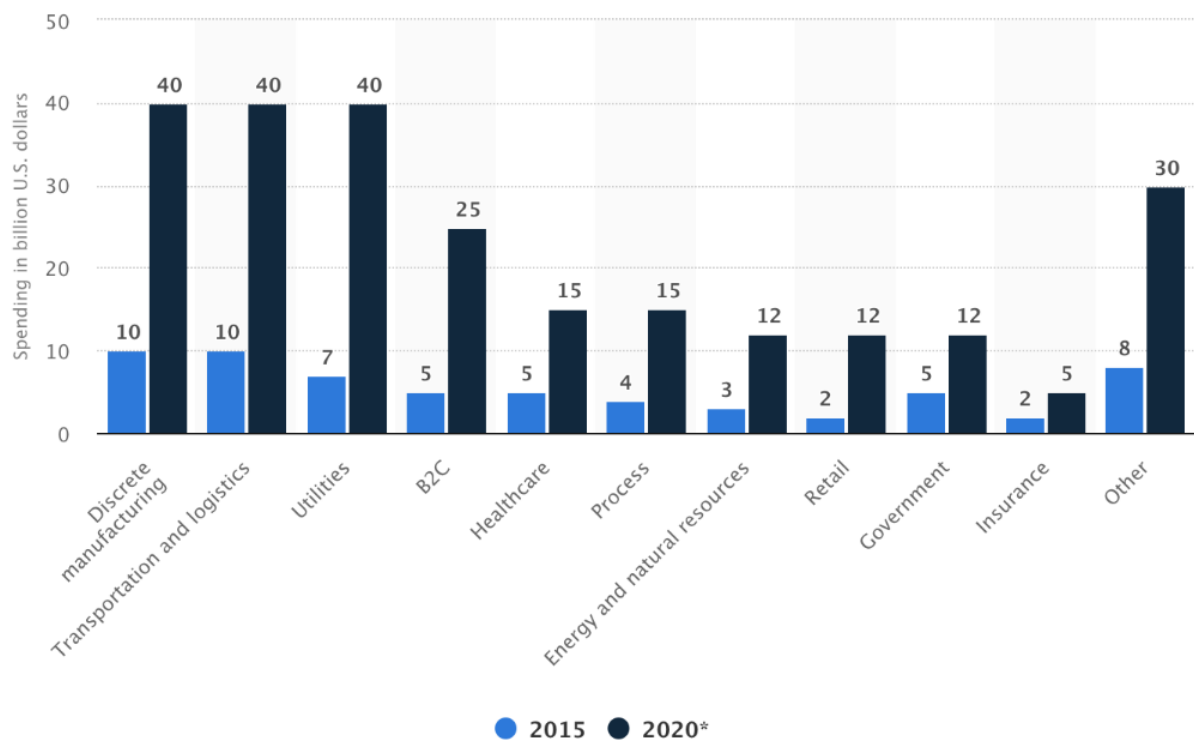
Também sob a perspectiva da consultoria de Gartner e IDC (*International Data Corporation*), a expansão da IoT será resultado da uma adoção cada vez maior por parte dos consumidores, estando imersos em objetos como (*smart watches*, sensores e atuadores domésticos para conforto ambiental e segurança. No âmbito dos negócios das mais diferentes matrizes (sincronização da cadeia de suprimentos, centros de fabricação que emitem alertas de manutenção, georreferenciação, dentre outras ferramentas) e no setor público (*smart cities*, semáforos, iluminação inteligente, transporte público, controle de inundações, distribuição de eletricidade, gás, águas, e coleta de lixo e de esgoto) também tem-se uma curva ascendente nas perspectivas. A IDC prevê um crescimento anual de 13,6% até 2022, quando o mercado de IoT atingirá US\$ 1,2 trilhão (BRITO, 2019).

A Figura 6 mostra o crescimento de investimento em bilhões de dólares nos diferentes setores da indústria, comparando dados de 2015 a 2022. As estatísticas mostram como os

gastos com a Internet das Coisas aconteceram de 2015 a 2020 em diferentes setores da indústria em todo o mundo.

É notável o crescimento rápido em investimentos nos setores de fabricação discreta (setor responsável pela produção de um produto em particular), *Utilities* (serviços de utilidade pública como energia, gás e água) e Transporte & Logística, quando comparamos com outros, atingindo expectativas de investimento de 40 bilhões de dólares em 2020. Nessa acepção, o aumento na velocidade, precisão e escala da cadeia de suprimentos é onde as organizações estão se concentrando, pois a IoT tem o potencial de reformular o gerenciamento da qualidade, rastreabilidade, conformidade e inteligência de fabricação.

Figura 6: Investimento em IoT nos mais diversos setores industriais



Fonte: (MITAL, 2018)

Dentre os principais desafios da IoT na atualidade:

1. Energia do sensor: Para que a IoT atinja seu potencial completo, os sensores deverão ser autossustentáveis e com desempenhos cada vez melhores. Por se tratarem de bilhões de dispositivos implantados no planeta (ou mesmo fora dele), qualquer necessidade de intervenção humana é inviável. Nesse sentido, propostas existem para que seja um mecanismo sustentável, como sensores para gerar eletricidade a partir de elementos ambientais, como vibrações, luz e fluxo de ar (SHWE; LIANG, 2009), ou mesmo nanogeradores comercialmente viáveis (PHYSORG, 2019).

Soluções como essa minimizam o impacto desse que é um dos maiores desafios atuais em IoT, sendo também trabalhado por topologias que minimizam o consumo de energia em determinada operação de comunicação entre nós.

2. Expectativa do usuário/cliente: quando tratado a nível de produto, a IoT enfrenta de certa forma um desafio na expectativa do cliente final. No passado, o tipo de tecnologia sofisticada que vemos hoje em residências em todo o mundo era em grande parte propriedade de uma comunidade de entusiastas de tecnologia. Com a integração completa de dispositivos complexos, como smartphones, em nossas vidas cotidianas, o mercado atual de produtos IoT mudou drasticamente. Ao contrário dos aficionados de ontem que utilizam-se de gadgets por diversão, os usuários de hoje são menos experientes em tecnologia, menos pacientes e mais propensos a devolver um produto que não funciona da maneira esperada (THOMPkins, 2018).
3. Padrões: organizações como a IEEE ano a ano buscam aprimorar suas normas a fim de possibilitar o roteamento de pacotes de IPV6 por diversos tipos de redes. Mesmo com avanços nessas normas, ainda há muito o que evoluir, principalmente no que se refere às áreas de segurança, arquitetura e privacidade dos dados. Uma barreira que provavelmente será quebrada nos próximos anos.

Outros desafios podem ser comentados como o armazenamento, processamento e visualização de dados que, diante do crescente número de objetos e dados em tráfego, exigem que novas tecnologias surjam com o objetivo de lidar com a quantidade de informações. Soluções na nuvem ou mesmo que apresentem aprendizado de máquina são algumas das propostas, tendo como desafio também a entrega dessas visualizações para o usuários finais (ČOLAKOVIĆ; HADZIALIC, 2018).

Outros desafios como a performance da rede, qualidade do serviço e segurança dos dados também são importantes e são considerados nos projetos atuais e perspectivas para a área da IoT.

2.2 Protocolos e Camadas em IoT

Para que objetos ditos inteligentes possam exercer a sua capacidade sistêmica é necessário que exista a conectividade entre os elementos. Descrevendo o modelo básico de camada e por ser o protocolo de comunicação muito aplicado, a IoT usa como base os conceitos de TCP/IP (*Transmission Control Protocol/Internet Protocol*, utilizado no seu início para protocolos de internet).

2.2.0.1 TCP/IP

Criada 1982, dentro do projeto americano ARPANET(*Advanced Research Projects Agency Network*), a pilha de protocolo TCP/IP tinha como objetivo a troca de dados entre

os sistemas situados em diferentes geolocalizações com certa tolerância a falhas de enlaces (BIDGOLI, 2004). Até hoje utilizada - na versão 4 da pilha - a topologia de protocolos possui quatro camadas no modelo: física/enlace, rede, transporte e aplicação.

A camada física consiste na comunicação de fato entre nós, sendo por onde passam os *bits* que compõem um dado. Seus protocolos são dependentes do protocolo de enlace, da forma como os *bits* são enviados e o meio físico em questão.

A camada de enlace gerencia a tecnologia de transmissão entre dois nós. No caso da IoT, a comunicação na camada de enlace se baseia em comunicação sem fio e, nesse sentido, diversos padrões foram especificados para a camada, no qual podemos citar os protocolos *Ethernet* (IEEE 802.3), o ATM (*Asynchronous Transfer Mode*), o PPP (*Point-to-Point Protocol*), *bluetooth* (IEEE 802.15.1) e o *WiFi*(IEEE 802.11) (SUTARIA; GOVINDACHARI, 2013).

Nesse sentido, há nessa camada a preocupação apenas com a transmissão entre nós, sem considerar critérios de decisão envolvendo quais enlaces serão tomados na transmissão efetiva. Fica a critério da camada de rede a responsabilidade de entrega de pacotes onde eles são realmente necessários (TANENBAUM, 2010). Utilizando o protocolo IP para atingir suas metas, há aqui a ideia de entregar, endereçar e evitar congestionamento - obstáculos - entre os nós.

A camada de transporte é responsável pela forma como os dados são enviados na rede. TCP e UDP (*User Datagram Protocol*) são os principais protocolos, que têm como finalidade permitir que as entidades partam da origem ao seus destinos, mantendo uma conexão (TANENBAUM, 2010). Dentre as diferenças do TCP para UDP, temos o primeiro realizando um controle grande de fluxo, ajustando a velocidade e congestionamento na transmissão garantindo, assim, melhor taxa de entrega. Diferente do que ocorre com o UDP, que não possui essas características de controle porém, em contrapartida, exerce o tempo de execução em menor tempo, sendo útil em cenários no qual a perda é aceitável.

A camada de aplicação refere-se ao nível no qual programas de computador estão presentes e exercem a troca de informações pela rede, além de ser local no qual estão presentes os protocolos de comunicação como HTTP, SSL, DNS, entre outros. (FONTE). Sendo o mais utilizado em serviços web, o HTTP possui alto consumo de energia quando refere-se a IoT.

A camada de transporte na arquitetura TCP/IP fornece controle de congestionamento e entrega confiável, ambos implementados pelo TCP, o protocolo dominante da camada de transporte na Internet. O TCP foi projetado por muitos anos para fornecer eficientemente uma grande quantidade de dados em uma conexão ponto-a-ponto de longa duração sem o requisito de latência rigorosa. Ele modela a comunicação como um fluxo de bytes entre o remetente e o receptor e impõe a entrega confiável em ordem de cada byte

no fluxo.

Dentre os principais motivos pelo não uso do protocolo TCP em IoT, podem ser citados (SHANG et al., 2016):

1. Restrições de energia: causado pelas situações de restrição de energia, tornando inviável manter uma conexão de longa duração em aplicativos IoT.
2. Baixa latência: em algumas aplicações há o requisito de baixa latência, o que pode não tolerar o atraso causado pelo *handshaking* TCP - sinais enviados entre dois dispositivos assegurando que a conexão entre ambos foi feita, que a velocidade e os protocolos estão corretos e, mais tarde, que os dados foram enviados e aceitos. .
3. Estabelecimento de conexão: boa parte de uma comunicação em IoT envolve apenas uma pequena quantidade de dados, tornando a sobrecarga de estabelecer uma conexão inaceitável.

Devido ao seu baixo custo, os dispositivos de computação IoT geralmente não são tão poderosos quanto os *desktops* tradicionais. A maioria dos dispositivos IoT tem pouca energia, usam um microcontrolador de baixo custo e memória limitada. Há aqui um desafio, já que os protocolos de Internet existentes normalmente não são projetados para esses dispositivos incorporados.

Nesse sentido, vários grupos de trabalho da Força-Tarefa de Engenharia da Internet (IETF - *Internet Engineering Task Force*) foram criados para lidar com esses problemas. O trabalho de padronização da IETF na IoT desempenhou um papel vital no estabelecimento dos protocolos de comunicação leves para ambientes restritos sobre a rede IP existente.

Estes incluem IPv6 em redes de área pessoal sem fio de baixa potência (6LoWPAN - RFC 6282) (J.; P, 2011), protocolo de roteamento IPv6 para redes de baixa potência e com perdas (RPL - RFC 6550) (BRANDT; HUI; KELSEY, 2012) e restrições Protocolo de Aplicação (CoAP: RFC 7252) (SHELBY; HARTKE; BORMANN, 2014). A Figura 7 mostra a comparação entre IETF IoT e Pilhas de protocolo TCP/IP.

2.2.1 Protocolos da camada de Aplicação em IoT

Responsável pela comunicação direta entre humanos e objetos conectados, a camada de aplicação realiza o gerenciamento global dos dados. Seus dois principais protocolos são o MQTT (*Message Queuing Telemetry Transport*) e CoAP (*Constrained Application Protocol*),

2.2.1.1 MQTT

Sendo um protocolo essencialmente voltado a mensagens e enviadas a um endereço denominado *topic*, o MQTT foi desenvolvido pela IBM com o objetivo de realizar a

Figura 7: Protocolos de rede IoT mapeados no modelo TCP/IP

Modelo TCP/IP	Protocolos em IOT
Aplicação	HTTPS, XMPP, CoAP, MQTT, AMQP
Transporte	UDP, TCP
Rede	IPv6, 6LoWPAN, RPL
Física & Enlace	IEEE 802.15.4 Wifi (802.11 a/b/g/n) Ethernet (802.3) GSM, CDMA, LTE

Fonte: Autor

comunicação entre satélites e equipamentos de extração de óleo. Dentre seus requisitos alinhados aos objetivos de uma implementação IoT estão o baixo consumo e a confiabilidade.

O protocolo possui arquitetura *server/customer* (servidor/cliente) com estilo de comunicação *publish/subscribe* (publicar/escrever), tendo todo dispositivo como cliente, o qual se conecta a um servidor e é denominado de *broker* (corretor).

Dentre as suas vantagens estão a eficiência energética do modelo, além de ser escalável. Além disso, por utilizar o protocolo TPC, possui elementos de segurança na rede. Por necessitar de um *broaker*, o MQTT se torna uma boa opção para comunicação remota/*cloud*, pelo fato do servidor *cloud* atuar como o *broaker* entre o dispositivo IoT e outros serviços.

2.2.1.2 CoAP

O protocolo CoAP tem como base a sua comunicação "um-para-um" que, diferente do protocolo MQTT com sua comunicação "muitos-para-muitos", possui suas informações sendo transferidas diretamente entre cliente e servidor. Nessa linha, seu foco está na interoperabilidade com a web.

De certa forma possui similaridades ao HTTP, porém possui pacotes infinitamente menores - lembrando sempre que dispositivos de IoT operam com energias limitadas.

Em relação as suas vantagens, possui menor consumo computacional e menor tempo de resposta quando acionado, pois mantém uma conexão ativa entre seus nós.

2.2.2 Protocolos da camada de rede

2.2.2.1 IPv6

IPv6 ou *Internet Protocol Version 6* é o nome dado ao protocolo que contém informações de endereçamento e controle para rotear pacotes que e execuções adaptadas a realidade da atual geração da internet ([TANENBAUM, 2010](#)). Ele foi projetado para fornecer várias vantagens em relação aos atuais cabos IP versão 4 (ou IPv4).

Ambos os protocolos definem o protocolo de camada de rede, ou seja, como os dados são enviados de um computador para outro computador através de redes de comutação de pacotes, como a Internet.

O IPv6 está documentado em vários RFCs a partir da RFC 2460. Embora o IPv6 seja o sucessor do IPv4, as duas versões de protocolo continuarão a ser protocolos orientados a dados para a Internet nos próximos anos ([DEERING; HINDEN; NOKIA, 1998](#)).

Quando tratadas as diferenças entre o modelo e seu antecessor, observa-se que o Ipv6 contorna o principal problema do IPv4, ou seja, o esgotamento de endereços para conectar uma rede comutada por pacotes. O IPv6 tem um espaço de endereçamento muito grande e consiste em 128 bits em comparação com 32 bits no IPv4. Portanto, agora é possível suportar um número muito grande endereços de IP exclusivos, um grande aumento no número de computadores que podem ser endereçados com o apoio do esquema de endereçamento IPv6 ([GAI, 1997](#)).

Dentre as principais melhorias desse protocolo, podem ser citados:

- Maior espaço de endereçamento
- Roteamento mais eficiente
- Melhor suporte de mobilidade
- Segurança
- Formato de cabeçalho simplificado para otimização de entrega de pacote
- Implementações para qualidade de serviço (QoS)

Essa grande quantidade de endereços possibilita que todos os equipamentos, dentre eles computadores, telefones IP, televisores digitais, possam ter endereços únicos globais, o que possibilitaria a alcançabilidade fim a fim de tais equipamentos, sem a necessidade

de processamentos especiais. Essa característica também pode ser citada como uma das melhorias, entrando em debate o conceito de QoS (*Quality of Service*).

O QoS é tratado em IPv6 da mesma maneira que é tratado em IPv4, possuindo suporte por classe de serviço através do campo de tráfego e do modelo DiffServ - Differentiated Services. Entretanto, o cabeçalho IPv6 tem um novo campo chamado flow label, que pode conter um rótulo identificando um fluxo específico de dados. Desta forma, o nó fonte gera uma rota de fluxo com rótulo, disponibilizando QoS nesse caminho, onde cada roteador do caminho toma ações baseadas por esse rótulo.

Superando o fator do aumento no número de bits do endereço IP que resulta no crescimento significativo do cabeçalho, o IPv6 tem o cabeçalhos menores quando comparado ao IPv4.

2.2.2.2 6LoWPAN

Definido pela IETF com o objetivo de tornar possível a operação da pilha IPv6 de forma suave e em ambientes que exijam esse comportamento, o 6LoWPAN (*IPv6 over Low power Wireless Personal Area Networks*) permite essa ação da pilha IPv6 na camada de enlace MAC IEEE 802.15 (J.; P, 2011)(OLSSON, 2014).

Para essa adaptação, esse protocolo comprime todos os cabeçalhos incluindo, 40 Bytes de cabeçalho IPv6 da rede e 8 bytes de cabeçalho UDP do transporte para alguns bytes. Dessa forma, para que quadros IPv6 sejam transmitidos por links de rádio IEEE 802.15.4, os quadros IPv6 devem ser dividido em partições e mais dados gerados para se parecer com o formato original.

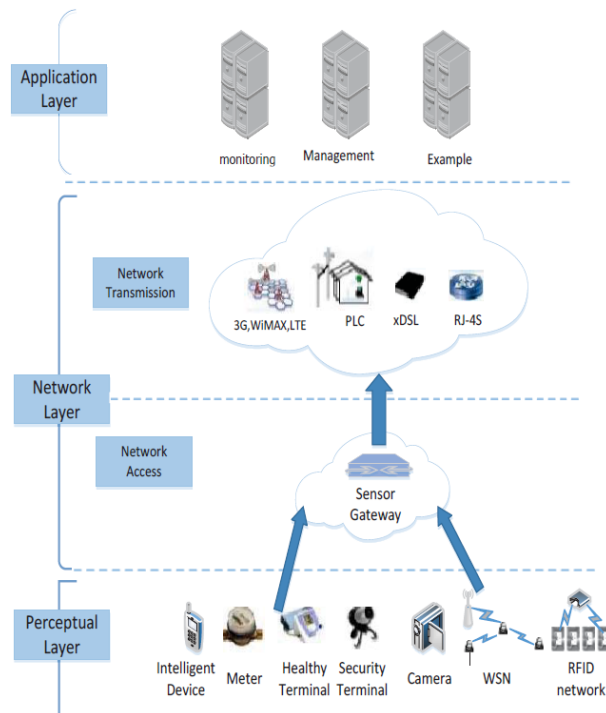
Existem três tipos de redes 6LoWPAN, dentre elas temos as *Ad-Hocs*, redes que não estão conectadas à Internet e que funcionam sem qualquer tipo de infra-estrutura. Além desse modelo, podem ser citadas as redes *Simple* e *Extended*, sendo redes inteligidadas a outras redes através de um ponto coordenador e redes com arquiteturas simples e ligadas por múltiplos coordenadores, respectivamente (J.; P, 2011).

2.3 Protocolo RPL

O protocolo RPL é um protocolo desenvolvido para aplicações focadas em em redes baixa potência e com perdas, as LLN's. (BRANDT; HUI; KELSEY, 2012).

Concretamente desenvolvido para redes LLN's, baseadas em IPv6, devido às suas deficiência na capacidade de energia e dos dispositivos em hardware, o protocolo de roteamento de vetor de distância atende aos dois requisitos bases: baixa taxa de dados e altas baixas de erros (GADDOUR; KOUBÂA, 2012). Surge aqui o conceito da DAG (do inglês, *Direct Acyclic Graphs*), que constitui um grafo direcionado que não possui ciclos.

Figura 8: Camadas e Arquitetura IoT



Fonte: (SHANG et al., 2016)

No protocolo RPL, a DAG permite que um nó se associe a mais de um nível, o que distingue o RPL dos demais protocolos de roteamentos baseados nessa topologia. A rede formada nesse cenário utiliza o conceito topológico de grafos acíclicos direcionados a destino, denominado DODAG (do inglês, *Destination oriented directed acyclic graphs*).

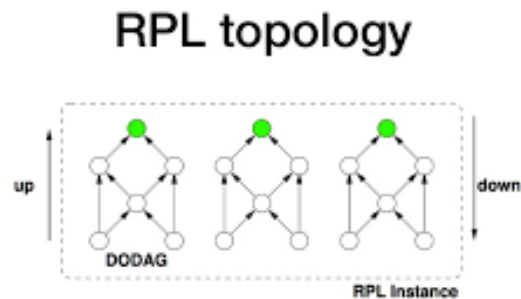
O comportamento de orientação de fluxo ocorre quando um dos nós assume a posição de pai, recebendo as informações dos nós filhos ele associados. Comumente chamado de nós raiz, o nó pai (também chamado de coletor) torna-se o topo da cadeia hierárquica e será responsável por enviar mensagens de controle e concentrar as informações recebidas pelos nós adjacentes (GADDOUR; KOUBÂA, 2012).

O grafo é criado por meio de mensagens de controle trocadas entre os nós e, após sua criação, este grafo pode definir os caminhos em que os dados podem percorrer e a constituição da topologia atual na rede. Essa rede pode ser constituída por um ou mais DODAGs e possui por esse motivo um valor identificador único chamado RPLInstanceID. Na figura 9 é possível observar a representação dessa instância, onde cada grafo é uma DODAG e há a comunicação entre os pontos de forma ascendente ou descendentes.

Assim que um nó RPL obtém o seu endereço IPv6 global logo une-se a um DODAG pronto e um nó pode participar de várias instâncias, porém de apenas um DODAG (BRANDT; HUI; KELSEY, 2012).

A Figura 9 mostra a direção da seta ascendente indica a comunicação de um nó

Figura 9: Instância de uma RPL



Fonte: (PISTER et al., 2009)

filho para seu nó raiz.

O RPL possui quatro valores de instância: ID de instância, ID do DOGAG, número da versão DODAG e *rank*. Tais valores são importantes para manter uma topologia DODAG e, assim, qualquer nó dentro da RPL pode ser identificado exclusivamente por esses 4 valores.

Em relação a instância, esse dado é importante na identificação dos DODAGs e, nesse sentido, nós que estão conectados a mesma raiz possuem o mesmo ID DOGAG, que é o que identifica o grafo orientado. O número da versão do DODAG é atualizado conforme a topologia das alterações do DODAG.

O *Rank* do nó refere-se a distância relativa do nó para a sua raiz e é um dos principais parâmetros para nós em RPL. Nós com ranks menores indicam que eles estão mais perto da raiz. Por ser um protocolo hierárquico cada nó seleciona para si um pai preferido, que obrigatoriamente possui um valor de rank menor que o seu, encaminhando posteriormente os dados.

Uma característica importante para o RPL é a possibilidade que o roteamento também seja feito entre nós de mesmo nível, combinando as topologias hierárquicas e *mesh*, provendo flexibilidade e gerenciamento entre as topologias (SCHRICKTE et al., 2013).

2.3.1 Mensagens de Controle

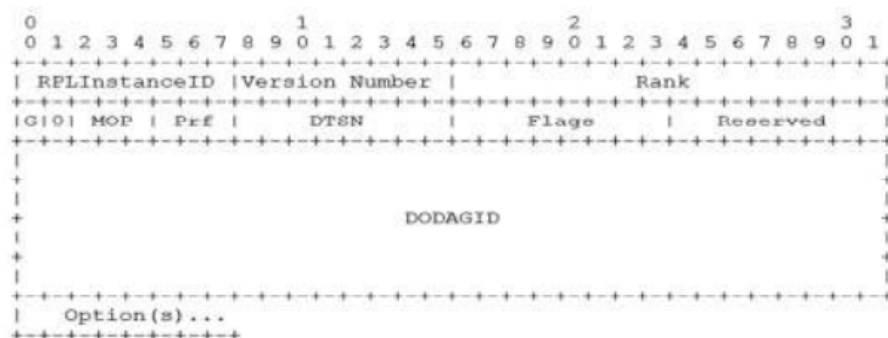
Durante a formação da rede, o RPL cria uma topologia em árvore, com seu roteador de borda servindo como raiz e seus nós descendentes formando uma cadeia de propagação de informações, seja ascendente ou descendente. Nesse sentido, cada nó da RPL possui sua classificação utilizando a função objetivo no controle desse dado. Ao longo desse seleção dos nós pais, espera-se que alguns mecanismos externos sejam acionados para controle

dessa acessibilidade dos pais e na disseminação das mensagens de controle. No acesso aos pais é utilizado o algoritmo *Neighbour Discovery* (ND), enquanto a mensagem de controle é disseminada pelo algoritmo *tickle time*.

Responsável por carregar informações necessárias para que um nó descubra uma instância RPL, aprendendo seus parâmetros, selecionando seus pais e conservando o DODAG, a DIO é a mensagem utilizada na criação desse novo DODAG. As informações contidas na mensagem DIO incluem os seguintes parâmetros: (BRANDT; HUI; KELSEY, 2012).

1. *RPLInstanceID*: valor de 8 bits único usado para identificar redes que compartilham a mesma função objetivo
2. DODAGID: identificador único dentro da Instância RPL, usado na identificação do DODAG raiz. O DODAGID é o endereço IPv6 roteável pertencente ao DODAG raiz.
3. *Version Number* - um contador seqüencial de 8 bits, incrementado sempre que o DODAG for reconstruído. Muito importante ao manter todos os nós sincronizados.
4. *Rank*: identificador de 16 bits da posição dos nós em relação aos outros nós da rede. Mesmo que o valor possa ser influenciado pela métrica do caminho, seu cálculo depende de suas próprias propriedades separadas seja para verificar a progressão do nó vinda ou indo a raiz. O *rank* do nó raiz é zero e aumenta pelo incremento mínimo de *rank* de 256 para baixo da raiz.

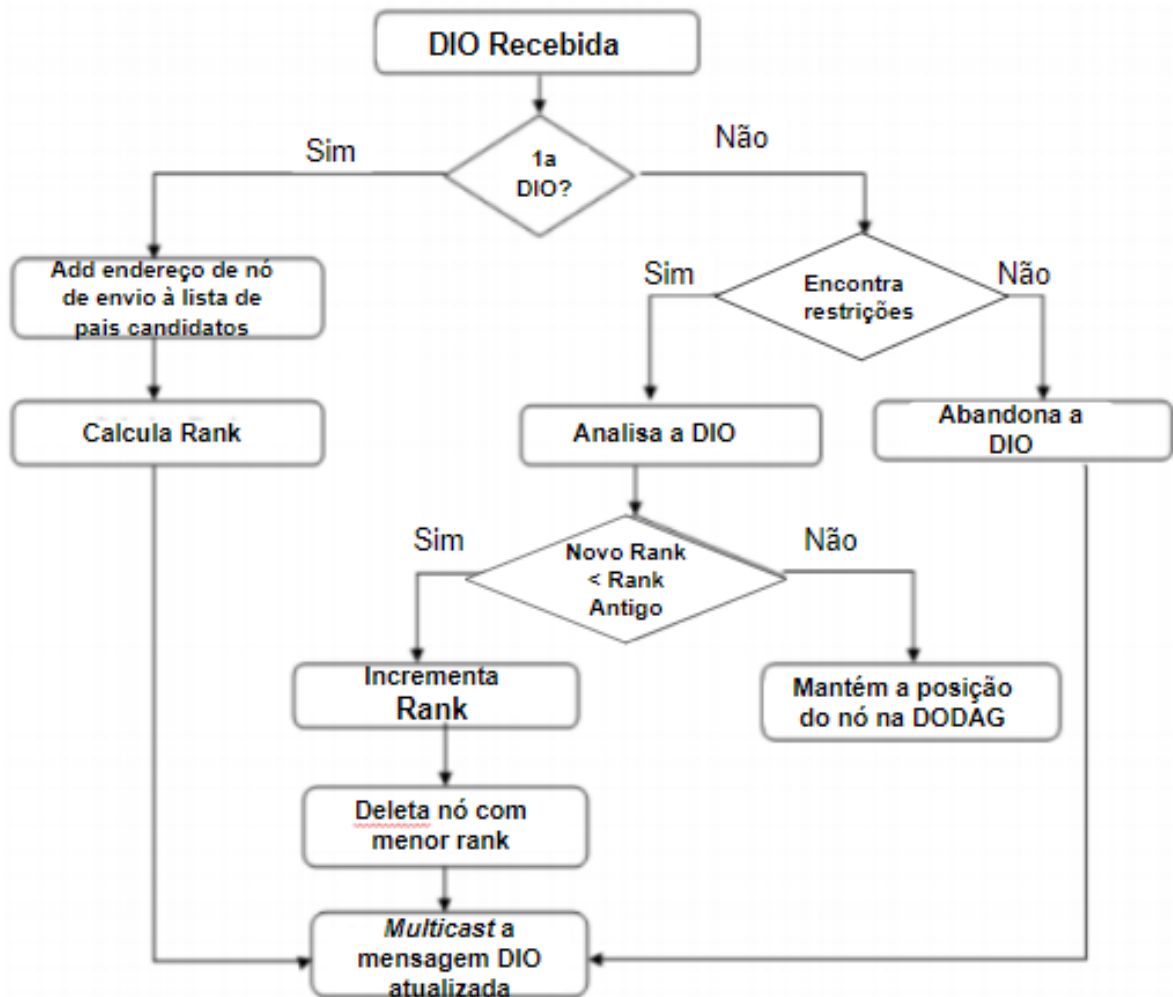
Figura 10: Cabeçalho mensagem DIO



Fonte: (BRANDT; HUI; KELSEY, 2012)

Como visto na Figura 10, o cabeçalho da mensagem DIO apresenta outros elementos como a *flag G* (*Grounded*) de 1 bit utilizado para indicar se o DODAG atual atende aos objetivos definidos na aplicação. O campo MOP (*Mode of Operation*) identifica o modo de operação definido pelo nó raiz. Nesse sentido, é um dado que precisa ser respeitado por todo novo nó.

Figura 11: Operação após recebimento da DIO



Fonte: Adaptado de (LAMAAZI; BENAMAR; J., 2017)

2.3.2 Criação da DODAG

O DODAG é o elemento principal da topologia RPL. Dessa forma, dentro de um DODAG cada nó recebe um *rank* indicando sua distância em relação ao nó principal - nó raiz. Assim, quanto maior o *rank*, mais longe o ponto está da raiz. Como apresentado, o *rank* é definido tomando como base o algoritmo apresentado na função objetivo em questão, guardando para si a informação de classificação própria e dos seus vizinhos.

Uma das principais capacidades da rede RPL é a de mudar rapidamente de pai preferido, principalmente quando existem nós móveis na rede. A partir do momento em que um nó sai da rede de alcance, este deixa de servir como rota em direção ao destino da informação.

Iniciando a construção da DODAG, o nó raiz tem a função de enviar uma mensagem de controle DIO a todos seus nós vizinhos, difundindo informações de *rank* desses pontos,

além a função objetivo utilizada na topologia. Outros detalhes como métricas e restrições da rede são enviados no pacote. A figura 11 mostra a operação após o recebimento de uma DIO por parte de um nó. Esse processo de difusão se repete até que todos os nós da malha sejam atingidos.

Um elemento importante na rede é o algoritmo *trickle timer*, capaz de restringir a sobrecarga de envios na rede, usando como base uma variação de tempo mínima na qual as mensagens podem ser enviadas. Esse tempo entre mensagens DIO cresce exponencialmente enquanto nenhum nó reporta um evento.

Iniciando a rota reversa, uma mensagem DAO é enviada ao nó pai. Assim que todos os nós receberem a mensagem de controle DAO, a DODAG estará completa.

A mensagem DIS é enviada quando algum novo nó queira ingressar e, ao receberem essa mensagem, os nós presentes na DAG retornam com uma DIO indicando o *rank* e as informações da malha.

2.3.3 Funções objetivo

Em RPL, a seleção do pai preferido está relacionado diretamente com a função objetivo que, além dessa importância possui a função de traduzir as métricas e restrições de uma rede em um valor de *rank*. Esse elemento é de suma importância pois um nó pode ter mais de um pai e, especialmente em uma rede de alta densidade, essa escolha é essencial. Logo, a maneira como é escolhida a função objetivo tem um impacto muito significativo na topologia da rede e desempenho de uma RPL (BRACHMAN, 2013).

Com o objetivo de tornar possível o suporte e acompanhamento do grande número de aplicações em redes de baixa potência e com perdas (LLNs), tanto a nível do nó quanto a nível do link, métricas são projetadas para RPL (J et al., 2013). Na realidade tais métricas podem estar associadas ao nó, ao roteamento e ao caminho.

A identificação da OF é feita pelo campo OCP na mensagem DIO. As funções objetivo propostas pelo IETF apresentadas em (THUBERT; SYSTEMS, 2012) e (GNAWALI; LEVIS, 2012) definem algumas recomendações sobre suas implementações, porém deixando em aberto qual métrica utilizar necessariamente. Em (THUBERT; SYSTEMS, 2012) é descrita a Função Objetivo 0 (OF0), sem definir nenhuma métrica e restrição. Garante, nesse sentido, um mecanismo simples de decisão do nó: associar-se a nó pai de acordo com o menor *rank*. Esta OF representa um denominador comum e não especifica nenhuma métrica de roteamento, mesmo apresentando o critério de menor *rank* e representar o menor caminho.

Uma segunda OF é citada, agora em (GNAWALI; LEVIS, 2012), o MRHOF (Minimum Rank with Hysteresis Objective Function) é proposta e é uma função baseada em *containers*, que são cabeçalhos para métricas localizados no pacote ICMPv6 DIO. Assim,

é possível que em um pacote DIO especificando a função objetivo MRHOF vários tipos de métricas de roteamento sejam indicadas. Essa função tem como principal característica a utilização de uma faixa de *rank* na qual há a decisão do nó se troca ou não de pai preferido. A MRHOF também tem como objetivo prover a rota com menor custo para envio feito pelos nós.

2.3.4 Métricas de avaliação

Desempenhando um papel muito importante na performance da função objetivo de roteamento, a análise de métricas de avaliação são podem ser agrupadas em elementos principais que, com seus resultados contrastados, nos permitem distinguir qual função objetivo que pode desempenhar em um cenário específico. Nesse trabalho, são citadas as 5 principais e, posteriormente, será dado destaque àquelas que apresentam maior significância no projeto.

2.3.4.1 Número de saltos

A função objetivo do protocolo pode utilizar como base para seleção do melhor pai o número de saltos entre eles. Um baixo número de saltos (em inglês, *Hop count*) significa que o nó é adequado para ser considerado melhor pai, assim por diante até o nó raiz. O número de saltos apresenta, nesse sentido, o número de saltos entre nós e seus vizinhos em direção à raiz. Quando lida-se com aplicação em tempo real, é sempre positivo considerar o menor número possível de saltos até chegar ao destino da informação (GADDOUR; KOUBÂA; ABID, 2015).

2.3.4.2 Energia consumida

Considerado um elemento de suma importância em projetos em IoT, o consumo de energia pode ser considerado uma métrica importante na escolha do nó pai, selecionando a rota em direção ao nó raiz com base no valor de baixa energia. Reduzir a energia é uma das métricas importantes em que a função objetivo se baseia nos critérios de decisão (YUNIS; DUJOVNE, 2014).

Esse dado pode ser calculado com base na energia consumida pelo nó na transmissão, recepção, além dos dados de *CPU*, que representa o consumo de energia durante o modo de potência total e *LPM* representando o poder consumo durante o modo de baixa potência, indicado em

$$E(mJ) = \frac{(Trans * 19.5mA + Recep * 21.5mA + CPU * 1.8mA + LPM * 0.0545mA) * 3V}{32768} \quad (2.1)$$

e considerando *Trans* a execução da transmissão do dado e *Recep* a recepção do dado.

2.3.4.3 Contagem de Transmissão Esperada (ETX)

Refere-se a quantidade de transmissões necessárias para que a informação seja recebida com sucesso no nó raiz. Esse valor pode prover informações importantes de estabilidade e custos com a comunicação dos nós. Considerando redes com perdas, esse é um dos parâmetros mais eficazes para o roteamento. Em (COUTO; AGUAYO; BICKET J. AMD MORRIS, 2003) a métrica foi proposta para aumentar a eficiência de redes sem fio. Em um cenário de transmissão, a métrica estima o número médio de tentativas necessárias para se obter uma transmissão com sucesso em cada enlace.

Essa métrica pode ser obtida por meio da expressão:

$$ETX = \frac{1}{D_r * D_f}$$

O D_f representa a taxa de entrega de pacotes dos enlaces direto, no qual é a probabilidade de que um pacote de informações seja recebido pelo seu vizinho. Em contrapartida, D_r é a taxa de entrega reversa, que é a probabilidade medida de um pacote reconhecido ser recebido com sucesso (LAMAASI et al., 2015). O enlace direto é utilizado para o envio dos dados enquanto que os reconhecimentos positivos (ACK's) são transmitidos no enlace reverso. A probabilidade de sucesso de uma transferência de um pacote de dados e de seu respectivo ACK é $d_f \times d_r$, o (ETX) é dado pelo seu inverso.

Redes com baixo ETX apresentam boa estabilidade, menor taxa retransmissão de pacotes e baixo consumo de energia nos nós.

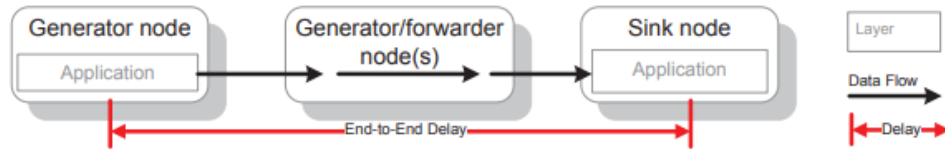
2.3.4.4 Perda de pacote

Na relações de comunicação entre nós, podemos utilizar a métrica de perda de pacotes de informação na comunicação entre esses elementos. Caracterizado como:

$$Pacotes\ Perdidos = \sum_1^n Pacotes\ Enviados - \sum_1^n Pacotes\ Recebidos$$

2.3.4.5 Delay End-to-End

A métrica *Delay end-to-end*, atraso de ponta a ponta, refere-se ao tempo gasto no caminho de um pacote desde a origem até o destino, considerando todo o tráfego na transmissão. Um termo usual no monitoramento de rede IP e O atraso de ponta a ponta ou o atraso unidirecional (OWD) refere-se ao tempo que um pacote demora a ser transmitido através de uma rede, da origem ao destino. É um termo comum no monitoramento de rede IP e difere do tempo de ida e volta (RTT) em que o único caminho em uma direção desde a origem até destino é medido.

Figura 12: Nós e *Delay End-to-end*

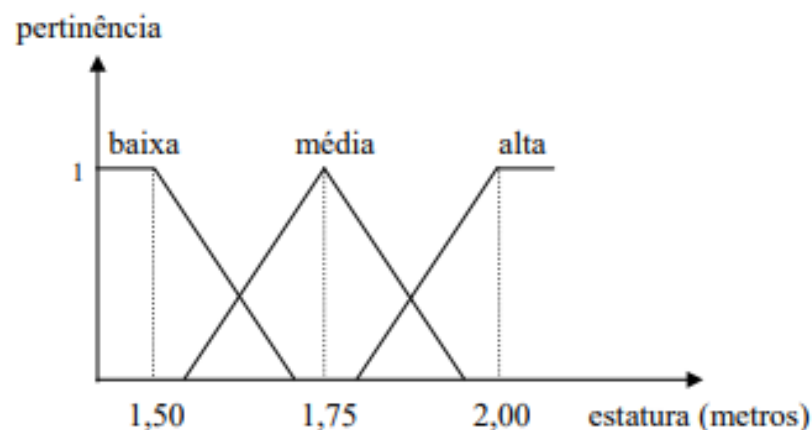
Fonte: d(PINTO; PINTO; RICARDO, 2013)

2.4 Lógica e Sistema *Fuzzy*

Conhecida como "A Teoria de Conjuntos *Fuzzy*" e concebida por L. A. Zadeh, a Teoria fuzzy tem como objetivo a disponibilização de um ferramental matemático para o tratamento de informações de caráter nebuloso ou vago. Com base nessa teoria, a Lógica Fuzzy foi caracterizada e seus operadores definidos à semelhança dos já utilizados tradicionalmente (L.A.ZADEH, 1965).

Diferente do comportamento da lógica clássica, na qual existem os conceitos consolidados de verdadeiro/falso (1/0), o avanço da tecnologia e das complexidades de cenários exigiram uma mudança na forma como os resultados podem ser encarados pela ciência. Há a definição de uma lógica que suporta os variados modos de raciocínio aproximado, no lugar de um raciocínio exato mais usual em qualquer vivência (CAVALCANTI et al., 2012).

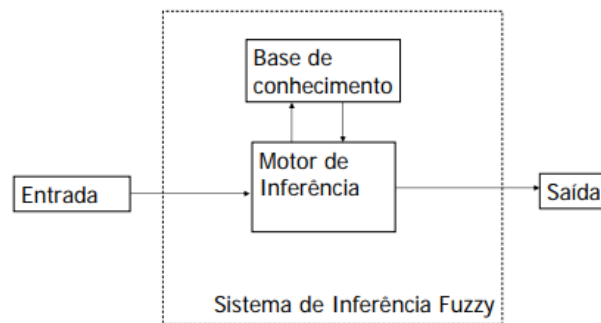
Figura 13: Função de pertinência



Fonte:

As primeiras noções da lógica dos conceitos nebulosos foram desenvolvidas pelo lógico polonês Jan Lukasiewicz (1878-1956) em 1920 que introduziu conjuntos com graus de pertinência sendo 0, $\frac{1}{2}$ e 1 e, mais tarde, expandiu para um número infinito de valores

Figura 14: Sistema de Inferência Fuzzy



Fonte: Autor

entre 0 e 1. A Teoria dos Conjuntos Fuzzy propõe algo diferente, generalizando a função característica de modo que ela, portanto, assuma sempre valores entre $[0,1]$.

Pode-se caracterizar um A (conjunto fuzzy) em um universo X como uma função de pertinência .

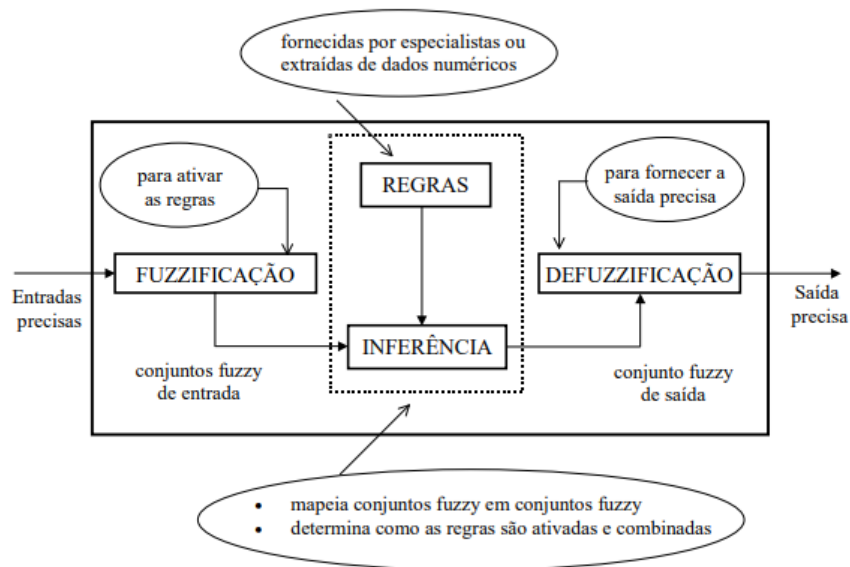
$$A = \{(x, \mu_a(x)) | x \in X\} \quad (2.2)$$

Nesse sentido, a função de pertinência de um universo envolvendo estatura, por exemplo, exibe valores de 0 a 1 com os diversos valores dessa variável. Seguindo o exemplo, podemos denominar a estatura como uma variável linguística e os pontos baixa, média e alta como sendo os valores dessa variável. Tais valores podem ser adaptados de acordo com o universo em análise: pouco, médio, muito, perto, longe, muito longe, dentre outros). Esse exemplo pode ser visto na Figura 13.

Há aqui também a definição de universo do discurso, adotando a escala de 0 a 2m e sentenciando o que é baixa, média e alta. Tais valores são calculados por meio do conceito e, logo, a função de pertinência é igual a 1 para o universo A (baixa) em estaturas de 0 a 1,5 metros. O mesmo vale num universo B, no qual a função é 1 para uma estatura 1,75 metros. Nesse cenário temos uma função pertinência triangular no universo B (média) e trapezoidal nos cenários A e C (baixa e alta), concluindo que uma pessoa com mais de 2,00 é certamente considerada alta no universo proposto.

Um Sistema de Inferência Fuzzy pode ser descrito como um processo no qual a Teoria de Conjuntos Fuzzy e Lógica Fuzzy fornecem o ferramental matemático para se lidar com as tais regras linguísticas. Neste sistema, como ilustrado na Figura 14, um bloco de entrada precisa resultados de observações, medições ou teorias que, passando por um processo denominado de motor de inferência (junto a base de conhecimento associada).

Figura 15: Sistema Completo de Inferência Fuzzy



Fonte:

Na Figura 15 pode-se verificar a etapa de *fuzzyficação*, esta responsável pelo cálculo do valor da função de pertinência da variável Fuzzy, mesmo processo mostrado na Figura 13. Há na *fuzzyficação* a avaliação do grau de pertinência da entrada numérica fornecida (X) gerando variáveis Fuzzy. Em seguida, há um processo de ativação das regras *fuzzy* formando um conjunto de saída, que será *defuzzyficado*. Dentro do bloco de inferência há o uso do das regras da base de conhecimento, como por exemplo ideias como "Se X1 é A2 e X2 é B1, então Y é C1".

No estágio de defuzzificação é efetuada uma interpretação dessa informação, sendo essencial em aplicações práticas, situações em que geralmente são requeridas saídas precisas. No caso de uma sistema de controle, por exemplo, em que é efetuado por um sistema de inferência fuzzy (ou controlador fuzzy), deve fornecer à planta dados ou sinais precisos, já que a "apresentação" de um conjunto fuzzy à entrada da planta não teria significado algum.

Dentre os diversos métodos de *deffuzzyficação* pode-se citar o método de centro de gravidade (CG), como visto na Equação 2.3. No modelo proposto, u é o valor consequente de saída.

$$u_{CG} = \frac{\sum_{i=1}^N u_i \cdot \mu_u(u_i)}{\sum_{i=1}^N \mu_u(u_i)} \quad (2.3)$$

2.5 Cenários de Mobilidade

Para que sejam introduzidos elementos de mobilidade, muitos cenários são possíveis e podem ser citados, desde humanos caminhando na rua com velocidade baixa e portando aparelhos sem fio, até mesmo drones sobrevoando uma cidade sem limitantes físicos como paredes ou ruas, enviando pacotes de dados a nós fixos.

Dentre as movimentações, pode-se citar também a rede móvel ad hoc (MANET, do inglês *Mobile Ad hoc NETWORK*), sendo definidas como uma rede de dispositivos móveis continuamente sem autoconfiguração e sem infraestrutura e conectada sem fio. As VANETs constituem um tipo específico de rede móvel ad hoc formada entre veículos (V2V) e entre veículos e dispositivos de infraestrutura (V2I), como pode ser visto na Figura 16 (LI; WANG, 2017).

Os dispositivos instalados nos veículos são conhecidos por unidades de bordo (OBU, do inglês *On-board unit*) e os que ficam ao longo da estrada são denominados unidades de acostamento (RSU, do inglês *Road-side unit*). Nesse envio de dados entre veículos, cada OBU funciona em modo ad hoc podendo encaminhar mensagens através de múltiplos saltos. Porém, nesse modo, a conectividade da rede é altamente dependente da quantidade de veículos na vizinhança e do padrão de mobilidade. Já modo V2I, a conectividade da VANET pode aumentar através de comunicação com outras redes, contudo, o custo de implantação também aumenta, pois há necessidade de termos RSUs espalhados pelas estradas e rodovias. (REHMAN et al., 2014).

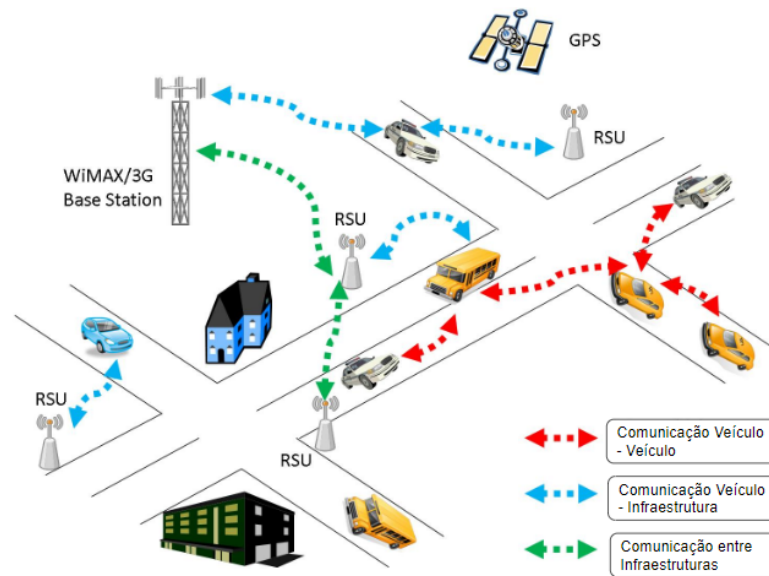
Por estarem imersos a regras de trânsitos, a movimentação ordenada dos carros faz com que nas VANETs exista uma previsibilidade na movimentação dos automóveis (REHMAN et al., 2014). Basicamente, uma VANET possui as seguintes características:

- Alta velocidade dos nós da rede (chegando a 150 km/h em rodovias, por exemplo);
- Fornecimento de energia limitado;
- Alta capacidade computacional;
- Movimentação previsível.

Tendo como objetivo fornecer diversas aplicações aos seus usuários, as VANETs operam tanto no que diz respeito à segurança quanto ao entretenimento. Operações envolvendo essas informações são conhecidas por Sistemas de Transportes Inteligentes (ITS, *Intelligent Transportation System*). Pode-se citar alguns exemplos de aplicações VANETs, como as aplicações de auxílio à mudança de faixa, aplicações de descoberta de melhor rota a um determinado destino, aplicações de divulgação de avisos de segurança, e acesso à internet, dentre outros (LI; WANG, 2017).

Genericamente, associa-se o uso intensivo de tecnologias da informação e da comunicação (TIC) às cidades inteligentes, com o objetivo de criar novas soluções aos problemas existentes, melhorando a eficiência dos serviços providos aos cidadãos. Por meio de diferentes sensores e recursos computacionais, automóveis podem recolher e tratar as informações com objetivo de auxiliar o usuário na tomada de decisões, somando forçar com elementos de comunicação sem fio como GPS (do inglês, Global Positioning System), por exemplo (REHMAN et al., 2014).

Figura 16: As redes ad-hoc veiculares (VANETs)



Fonte: Adaptado de (BEDI; JINDAL, 2014)

3 MATERIAIS E MÉTODOS

3.1 Sistema Operacional Contiki

Contiki é um sistema operacional de código aberto para IoT. Sendo considerado um SO (Sistema Operacional) leve para sistemas embarcados de rede, o Contiki que fornece mecanismos para desenvolvimento de softwares para IoT e mecanismos de comunicação que permitem a comunicação entre dispositivos inteligentes ([CONTIKI-OS, 2013](#)).

Além disso, disponibiliza bibliotecas para a alocação de memória, abstrações de comunicação e mecanismos de redes de rádios de baixa potência, sendo um poderoso bloco de ferramentas para construção de sistemas de rede sem fio complexos, além de projetos como esse envolvendo a mobilidade de nós em uma rede. Outro ponto é a possibilidade de conectar-se a pequenos microcontroladores de baixo custo e baixo consumo de energia à Internet.

Por ser projetado para execuções ocupando pouco espaço de memória, é altamente eficiente de memória. Possui rede IP completa, com protocolos como: UDP, TCP, HTTP, 6LoWPAN, RPL, COAP e IPv6. Opera em sistemas de baixo consumo de energia e seu sistema de compilação torna mais fácil: compilar aplicativos para qualquer uma das plataformas Contiki disponíveis e com isso facilita o teste de aplicativos em uma variedade de plataformas ([CONTIKI-OS, 2013](#)).

Dentre as principais características e vantagens do Contiki, tem-se ([CONTIKI-OS, 2013](#)):

1. Padrões de Internet: Disponibiliza uma eficiente comunicação de baixa potência na Internet, suportando os padrões IPv6 e IPv4, juntamente com os recentes padrões sem fio de baixo consumo de energia: 6lowpan, RPL, CoAP.
2. Agilidade no Desenvolvimento: utilizando o Contiki, o desenvolvimento torna-se mais rápido e prático que alguns dos seus similares: os aplicativos Contiki são gravados no padrão C, com o simulador Cooja. As redes Contiki podem ser emuladas antes de serem gravadas no hardware e o Instant Contiki fornece um ambiente de desenvolvimento inteiro em um único download.
3. Memória: O Contiki é projetado para ser executado em pequenas quantidades de memória. Um sistema típico com rede IPv6 completa com roteadores sonolentos e roteamento RPL precisa de menos de 10k de RAM e 30k de ROM.
4. Código aberto: Contiki é open source, o que significa que a fonte está e sempre estará disponível. O Contiki pode ser usado em sistemas comerciais e não comerciais sem

restrições.

5. *The Rime Stack*: Contiki fornece uma pilha de rede sem fio personalizada chamada *Rime*. Ela é utilizada em situações na qual a largura de banda é escassa ou onde a pilha de rede IPv6 é excessiva. A pilha personalizada suporta operações simples, como o envio de uma mensagem para todos os vizinhos ou para um vizinho especificado, bem como mecanismos mais complexos, como inundação de rede e coleta de dados escalonáveis.
6. Consciência de Potência: projetado para operações em redes e sistemas de baixíssima potência: sistemas que podem precisar funcionar durante anos com um par de baterias AA, por exemplo. Para auxiliar o desenvolvimento de sistemas de baixa potência, a Contiki fornece mecanismos para estimar o consumo de energia do sistema e para entender de forma autônoma onde há excesso de gasto de energia.
7. 6lowpan, RPL, CoAP: O Contiki suporta os protocolos IETF recentemente padronizados para rede IPv6 de baixa energia, incluindo a camada de adaptação do 6lowpan, o protocolo de roteamento de vários saltos RPL IPv6 e o protocolo de camada de aplicativo CoAP RESTful.

O sistema operacional é considerado o primeiro para objetos inteligentes a fornecer comunicação IP com a pilha IP TCP/IP e, por utilizar IP, pode comunicar-se diretamente com outros aplicativos baseados em IP. O sistema operacional e suas aplicações são implementados em linguagem C. (SANTOS et al., 2016).

O sistema operacional Contiki tem sua execução juntamente com o simulador Cooja, tornando possível a simulação dos comportamentos das aplicações advindas da rede IoT em seus diversos cenários.

3.2 Simulador Cooja

O simulador Cooja (*Contiki OS Java Simulator*) faz parte ao Contiki, sendo constituído de arquivos *.java* e *.class* basicamente, portanto um software *Java*. O simulador de instruções em cada nó da rede - denominado como *mote* - é o MSPSim, simulador baseado em Java para rede de sensores.

O código é compilado dentro da aplicação e, a partir da compilação, é gerado código objeto ao sistema correspondente.

Todo o sistema é controlado e analisado, possuindo uma interface para analisar e interagir com os nós, o que facilita o trabalho e a visualização da rede. Além disso, é possível criar cenários personalizados (SANTOS et al., 2016).

Tabela 1: Sistemas de Mote e Microcontrolador disponíveis no Simulador Cooja

Mote	Microcontrolador
Wismote	MSP430F5437
Z1	MSP430x2xx
Sky	MSP430F1611
ESB	MSP430
MicaZ	ATmega128
eth1120	MSP430
trxeb1120	MSP430F5438
trxeb2520	MSP430
Exp2420	MSP430F5438
EXP430F5438	MSP430F5438

Tabela 2: Fonte: Adaptado de (INCORPORATED, 2017)(INCORPORATED, 2002)

Tabela 3: Memória dos Microcontroladores disponíveis no Simulador Cooja

Microcontrolador	RAM	ROM	EEPROM	Flash
ATmega128	4 KB	-	4KB	128KB
MSP430	128 bytes	1KB	-	-
MSP430F1611	128 bytes	2KB	-	48KB
MSP430x2xx	256 bytes	4KB	-	56KB
MSP430F5437	512 bytes	16KB	-	256KB
MSP430F5438	512 bytes	16KB	-	256K

Tabela 4: Fonte: Adaptado de (INCORPORATED, 2017)(INCORPORATED, 2002)

O Cooja possui alguns modelos de propagação já implementados, que podem ser utilizados para simulações em diferentes ambientes, porém ainda existem muitos modelos teóricos que não possuem implementação.

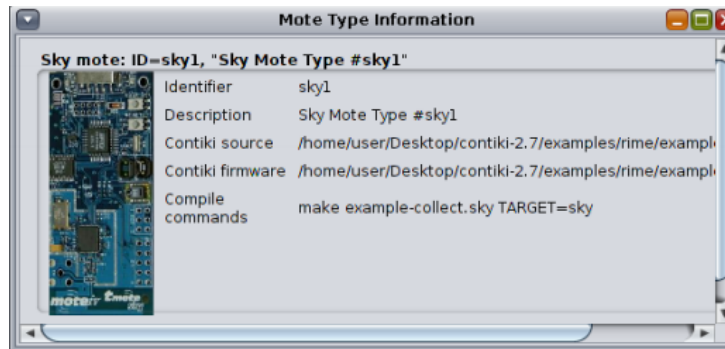
No que fiz respeito ao sistema de nós e microcontroladores disponíveis, temos a descrição técnica presente na tabela 1, além dos fatores envolvendo as memórias RAM, ROM, EEPROM e Flash, apresentados na tabela 4.

Presente na figura 17, é possível verificar os detalhes operacionais do micro utilizado nos nós transmissões e receptores do simulador, denominado *sink* mode. Nele é possível verificar a descrição do mote, com seu ID, tipo e identificador.

3.3 Simulador gerador de Mobilidade - Modelo *Manhattan*

Com o objetivo de contruir um cenário móvel para simulação das métricas de atenção, o Simulador Cooja com a implementação de mobilidade é utilizado.

Figura 17: Descrição do *sky mote* com *sky mote ID*, *sky mote type*, identificador, fonte contiki e *firmware*

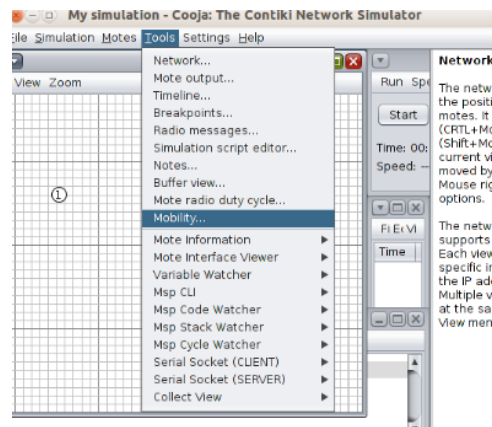


Fonte: Simulador Cooja

Essa ativação da plataforma para cenários móveis é realizada com o objetivo de testar e simular protocolos de rede ad hoc.pois de seguir todas as etapas mencionadas neste tutorial, é possível simular com sucesso nós móveis em seu simulador de Cooja (??).

Dentre as principais características do modelo *Manhattan Mobility* é possível notar: o nó móvel pode se mover ao longo da grade de ruas horizontais e verticais no mapa; em uma intersecção de uma rua horizontal e vertical, o nó móvel pode virar à esquerda, direita ou ir direto com certa probabilidade; com exceção da diferença acima, os relacionamentos entre nós e entre nós envolvidos no modelo de *Manhattan* são os mesmos do modelo *Freeway*.

Figura 18: Módulo de Mobilidade Implementado no Simulador Cooja

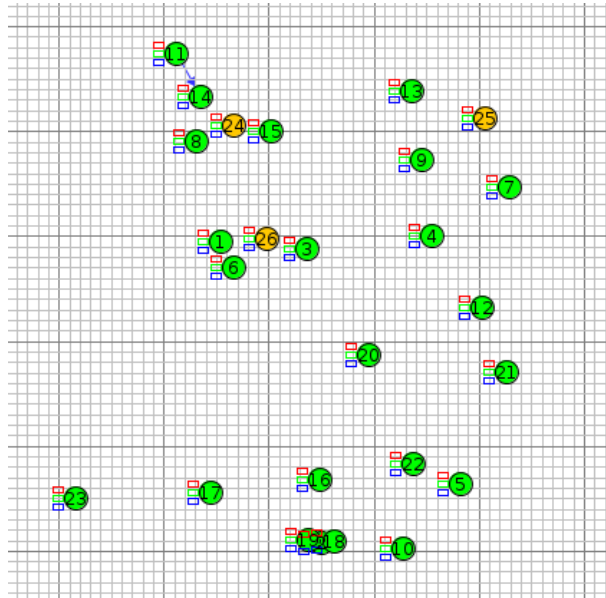


Fonte: Simulador Cooja

Nas figuras 19, 21 e 20 exibem a distribuição de nós fixos e móveis numa área de 500 metros x 500 metros. É importante destacar que trata-se de um *grid* padrão de 10 metros por 10 metros, simulando uma rede urbana de mobilidade.

Como pode ser visto na figura 20, os nós emitem pacotes em busca do melhor nó pai e, nos dados exibidos em um dos *frames*, é possível visualizar o sentido do envio e as posições dos nós mostrados na numeração.

Figura 19: Ambiente 500 metros quadrados contendo 25 nós



Fonte: Simulador Cooja

Nos experimentos são estruturados os três cenários de mobilidade:

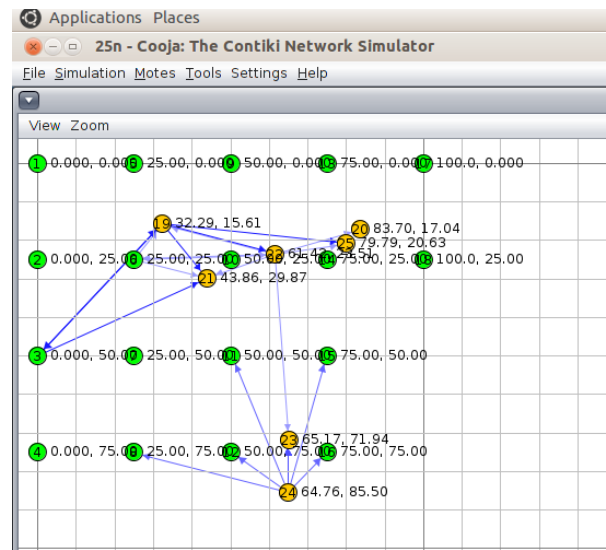
1. Cenário 1: 25 nós, sendo 20 nós fixos + 5 nós móveis
2. Cenário 2: 30 nós, sendo 20 nós fixos + 10 nós móveis
3. Cenário 3: 40 nós, sendo 20 nós fixos + 20 nós móveis

Na figura 21 é possível notar o raio de radiofrequência de um dos nós, mostrando como esse nós se conecta com um nó pai - ou quando é conectado por outro nó filho - e inicia a construção da DAG.

Então, o modelo de mobilidade de *Manhattan* tem alta dependências temporais e também impõem restrições geográficas na mobilidade dos nós, mas dar alguma liberdade aos nós como para mudar sua rua. O modelo de mobilidade de *Manhattan* é útil na modelagem movimento em áreas urbanas.

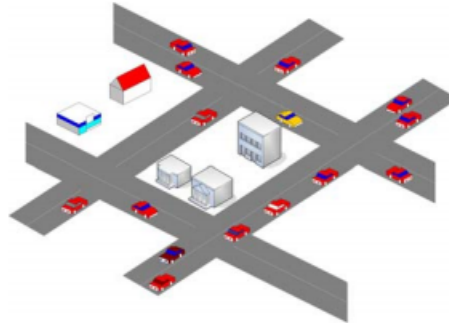
A aplicação do gerador de mobilidade é feito ao adicionarmos o o diretório `position.dat`, visto com mais detalhes na seção 5. Na figura 18 é possível notar o diretório disponível após a implementação do app de mobilidade.

Figura 20: Cenário com os nós enviado/recebendo pacotes e as posições indicadas de cada um dos sensores



Fonte: Simulador Cooja

Figura 22: Cenário de mobilidade Manhattan



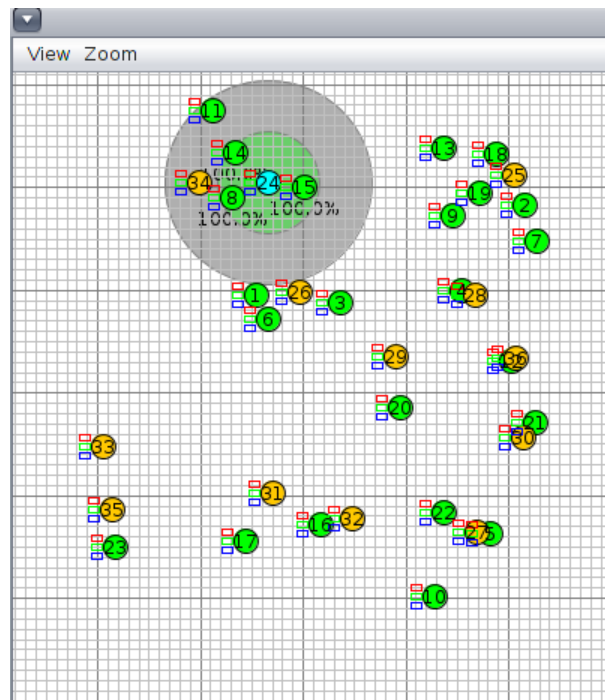
Fonte: (ALAM; SHER; HUSAIN, 2009)

3.4 Método proposto de Função Objetivo baseado em lógica fuzzy

Nessa trabalho é proposta uma melhoria do RPL com uso de uma função objetivo baseada na combinação de algumas das métricas citadas e utilizando uma abordagem com lógica fuzzy, reafirmando a importância da análise em ambientes dinâmicos, com mobilidade, o escopo deste trabalho.

Com o objetivo de implementar e simular no ambiente Cooja, denominamos as funções objetivos propostas como FuzzyC1, FuzzyC2 e FuzzyC3, analisando o impacto dessas novas modalidades de algoritmo de seleção de nó quando comparadas com as funções objetivos já consolidadas na literatura RPL: OF0 e MRHOF.

Figura 21: Ambiente de simulação com os nós e a área de abrangência do nó 24



Fonte: Simulador Cooja

Na proposta de OF, consideram-se as métricas *Delay end-to-end*, ETX e número de saltos no mapeamento dos comportamentos do protocolo e na construção das 3 OF's citadas. Tendo como objetivo a seleção do melhor nó pai em cada interação nó-nó vizinho em direção a raiz do DAG, cada uma dessas métricas descreve uma importante ligação e o conjunto de métricas é usado para avaliar essa qualidade do nó.

Para a decisão do pai preferido recorreremos à lógica fuzzy na implementação da OF. Com o objetivo de melhorar o desempenho da rede global e por se tratar de uma rede de sensores sem fio, pode-se descrever com o uma técnica de inteligência artificial para tomada de decisão e considerando que lidamos aqui com baixa potência aplicada.

Desde que pretendemos decidir sobre o melhor pai entre a lista vizinhos dependendo dos requisitos de aplicação em um maneira eficiente, nossa proposta faz uso da lógica fuzzy desempenha um papel importante na tomada de decisões.

Dentre as vantagens da lógica fuzzy podem ser citadas:

- Torna possível um raciocínio abstrato sobre qualquer faixa de valores;
- Fornece uma álgebra para lidar dados imprecisos;
- É um método eficiente de combinar objetivos conflitantes e base de conhecimento humana;

- Pode ser implementado com algoritmos de baixa complexidade.

Em resumo, serão aplicados conceitos de um sistema fuzzy, composto por um componente de *fuzzyficação*, capaz de traduzir elementos exatos de entrada - as métricas citadas, nesse caso - em valores fuzzy. Logo em seguida, será utilizado o componente de motor de inferência, capaz de utilizar a base de conhecimento sobre o universo em análise para que, por fim, exista uma *defuzzificação* capaz de gerar um dado único, a qualidade do nó na transmissão, ponderando as 3 saídas e decidindo, por fim, qual melhor caminho entre os nós.

3.4.1 Sistema Fuzzy

Apresentado no capítulo anterior, o sistema fuzzy tem como objetivo realizar, pelo cálculo de valores fuzzy provenientes do módulo de interface de entrada, o processamento, junto das regras existentes na base de conhecimento, de um conjunto fuzzy de saída para o módulo de interface correspondente, a partir da composição de todas as regras de entradas disparadas.

Por uma regra disparada, entende-se uma regra cujo processamento do antecedente para as entradas atuais gerou graus de pertinência não-nulos, ou seja, a relação fuzzy entre as entradas e os termos primários do antecedente é maior que zero.

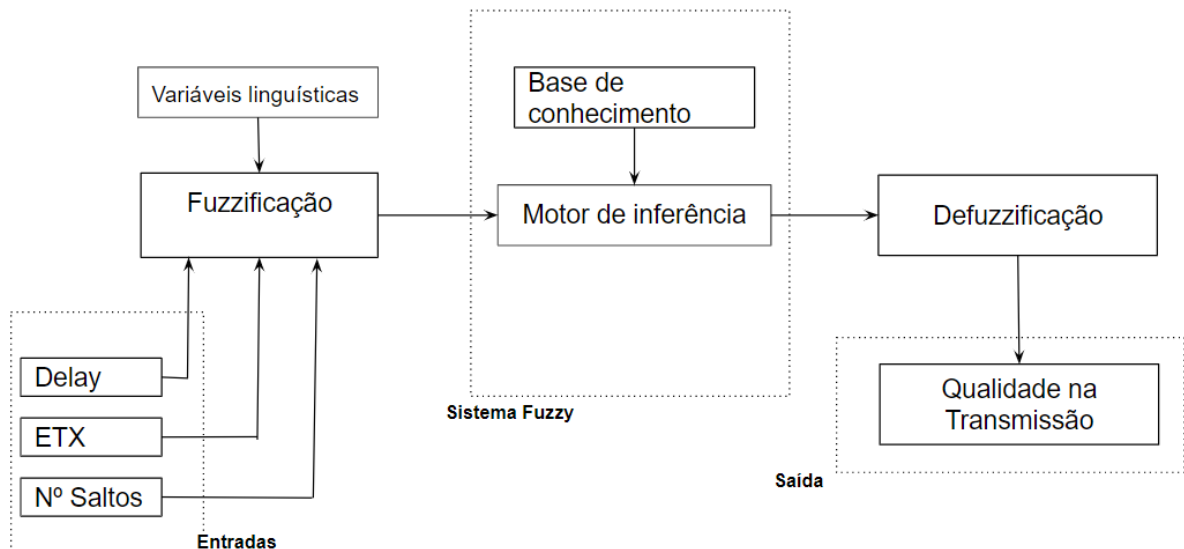
Na Figura 23 é possível identificar os seguintes módulos para o projeto: blocos de entrada, fuzzificação, variáveis linguísticas, motor de inferência, base de conhecimento, defuzzificação e a saída do processo fuzzy.

No Sistema de inferência fuzzy, regras são usadas para avaliar o sistema e dar significado ou resultado às diferentes combinações de condições de entrada que podem ser obtidas do sistema. Consideramos aqui os elementos de entrada as 3 métricas estratégicas: *Delay End-to-End*, ETX e o número de saltos.

O processo de fuzzificação mostrado na Figura 23 envolve a introdução dos valores da métrica de entrada do sistema, em seguida, determine sua função de pertinência para algum conjunto fuzzy definido. A avaliação da regra é então realizado para combinar diferentes valores de entrada para obter os conjuntos fuzzy de saída, que é finalmente defuzzificado para obter um único valor de métrica de saída, considerando as novas funções objetivos propostas em um móvel cenário de rede.

A base de regras presentes no motor de inferência são: variável de entrada, peso da variável de entrada (P), valor fuzzy definido, valor antecedente e, por fim, a saída do sistema (qualidade), mostrados com elementos do projeto nas seções seguintes.

Figura 23: Estrutura do processo Fuzzy



Fonte: Autor

3.4.2 Composição de métricas e Fuzzyficação

As métricas de entrada foram selecionadas levando-se em consideração à sua importância e relevância no universo de estudos da mobilidade do sensor sem fio redes. Cálculos e ajustes adequados destes parâmetros são necessários para alcançar um função objetiva otimizada para mobilidade em um cenário de comunicação sem fio RPL.

São escolhidas 3 métricas que, para o processo fuzzy, influenciam nos benefícios nos dados de roteamento em WSN móvel. Nesse sentido, há aqui uma limitação de memória inerente às redes de baixa potência sem fio o que não recomenda a expansão para mais entradas dentro do processo de decisão, o que ocasionaria um aumento significativo nas regras dentro do algoritmo.

Os parâmetros de entrada considerados no projeto são;

1. *Delay End-to-End*: essa métrica representa a soma de latências de link, que é definida em como um aditivo agregado métrica. Essa métrica deve ser minimizada para aplicativos que exigem garantias em tempo real e por isso a importância em estar aqui.
2. ETX (Contagem esperada de transmissões): como apresentado, é uma métrica de camada de enlace de software e impacta diretamente na quantidade de mensagens

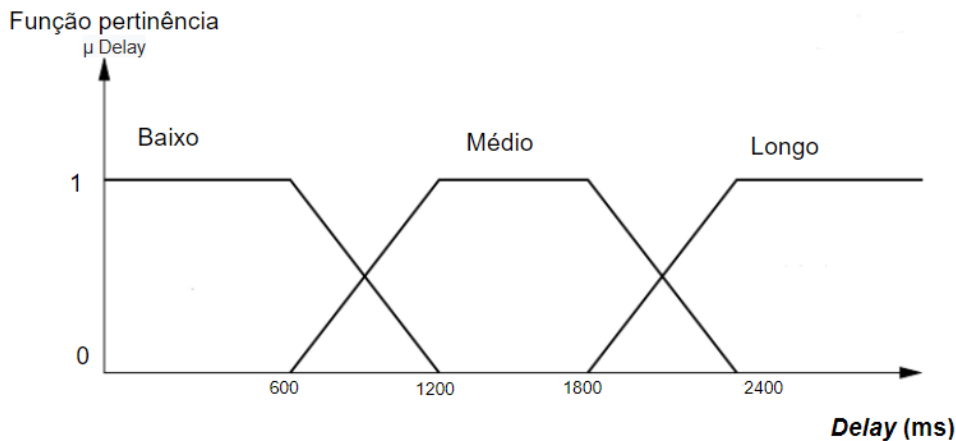
de controle geradas na DAG, garantindo que o caminho escolhido será sempre o que possui o menor número de retransmissões.

3. Número de Saltos: Esse *input* auxilia na seleção da menor contagem de saltos, como apresentado no capítulo anterior. Em uma situação na qual dois ou mais nós possuem o mesmo *delay* e mesmo ETX, o caminho (nó) com menos saltos será selecionado para rotear o pacote.

3.4.2.1 Métrica *Delay End-to-End*

Conforme visto na Figura 24, a função de pertinência para o atraso é um gráfico representativo dos graus de verdade de determinada variável linguística adotada no projeto. Para o parâmetro de entrada de atraso são considerados os graus "Baixo", "Médio" e "Longo".

Figura 24: Função Pertinência para *Delay*



Fonte: Autor

Por meio de simulações utilizando RPL no simulador Cooja em cenários aleatórios (*Random Waypoint model* e *Random Walk model*, nesse caso) e apoio teórico, como apresentado em (J et al., 2013) e (GADDOUR et al., 2012), pode-se concluir que o atraso de ponta a ponta em ambientes móveis está num intervalo até 4 segundos em média.

Considerando um ajuste nessa variação de tempo é possível determinar a função $Baixo(Delay)$ igual a 1 quando menor ou igual a 600 milissegundos, um $Medio(Delay)$ igual a 1 quando o atraso está entre 1200 e 1800 ms. A partir de 2400 ms pode-se concluir que estamos diante de um $Alto(Delay)$ igual a 1.

Nas equações 3.1 a 3.3 são apresentadas todas as equações que descrevem a função de pertinência *Delay* para as três variáveis linguísticas, adotando as regras de avaliação

dos intervalos do gráfico:

$$Baixo(Delay) = \begin{cases} 1 & \text{se } Delay \leq 600 \\ \frac{Delay-1200}{600-1200} & \text{se } 600 < Delay < 1200 \\ 0 & \text{se } Delay \geq 1200 \end{cases} \quad (3.1)$$

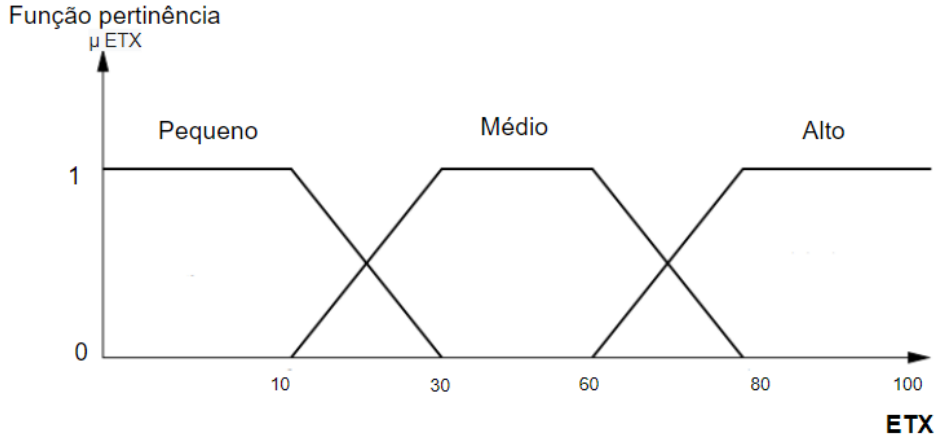
$$Medio(Delay) = \begin{cases} 0 & \text{se } Delay \leq 600 \\ \frac{Delay-600}{1200-600} & \text{se } 60 < Delay < 1200 \\ 1 & \text{se } 1200 < Delay < 1800 \\ \frac{Delay-2400}{1800-2400} & \text{se } 1800 < Delay < 2400 \\ 0 & \text{se } Delay \geq 2400 \end{cases} \quad (3.2)$$

$$Alto(Delay) = \begin{cases} 0 & \text{se } Delay \leq 1800 \\ \frac{Delay-1800}{2400-1800} & \text{se } 2400 < Delay < 1800 \\ 1 & \text{se } Delay \geq 2400 \end{cases} \quad (3.3)$$

3.4.2.2 Métrica ETX

Para enviar pacotes com sucesso ao destino é necessário o somatório do caminho ETX para a raiz. No arquivo *mrhof.c* presente na seção de Anexo B na seção 5, o valor máximo de retransmissões considerado, o MAX PATH COST, é definido como 100. Dessa forma, adota-se essa medida para o desenho das regras dos parâmetros de entrada do ETX variável.

Assim, o valor de caminho máximo de 100 é usado para mapear o valor de entrada para a função de associação em intervalo de $[0,1]$ como mostrado na figura abaixo.

Figura 25: Função Pertinência para ETX 

Fonte: Autor

O gráfico mostra o mapeamento do ETX do caminho para as variáveis linguísticas "Pequeno", "Médio" e "Grande".

Com base em estudos apresentando o comportamento do número esperado de transmissões (THOMSON et al., 2017) e também simulações em ambientes de mobilidade utilizando o simulador Cooja, o mapeamento da variável linguística foi feito para o ETX e as equações 3.4 a 3.6 exibem as funções de pertinência nas três variáveis: $Pequeno(caminhoETX)$, $Medio(caminhoETX)$ e $Alto(caminhoETX)$.

$$Pequeno(caminhoETX) = \begin{cases} 1 & \text{se } caminhoETX \leq 10 \\ \frac{caminhoETX-30}{10-30} & \text{se } 10 < caminhoETX < 30 \\ 0 & \text{se } caminhoETX \geq 30 \end{cases} \quad (3.4)$$

$$Medio(caminhoETX) = \begin{cases} 0 & \text{se } caminhoETX \leq 10 \\ \frac{caminhoETX-10}{30-10} & \text{se } 10 < caminhoETX < 30 \\ 1 & \text{se } 30 < caminhoETX < 60 \\ \frac{caminhoETX-80}{60-80} & \text{se } 60 < caminhoETX < 80 \\ 0 & \text{se } caminhoETX \geq 80 \end{cases} \quad (3.5)$$

$$Alto(caminhoETX) = \begin{cases} 0 & \text{se } caminhoETX \leq 60 \\ \frac{caminhoETX-60}{80-60} & \text{se } 60 < caminhoETX < 80 \\ 1 & \text{se } caminhoETX \geq 80 \end{cases} \quad (3.6)$$

Valores de 0 a 10 estão no grau de associação de 1 (valor booleano de *true*), dentro do nível linguístico variável *Pequeno(caminhoETX)*. Dessa forma, caso o valor seja acima de 10, mas menor que 30, seu grau de verdade deve ser calculado usando elementos de uma equação de reta de primeiro grau. A regra é seguida até atingirmos o valor 1 dentro da função *Alto(caminhoETX)* para valores maiores que 80 retransmissões.

3.4.2.3 Número de Saltos

Utilizando como base de conhecimento o Contiki RPL, essa métrica pode ser obtida por meio do parâmetro *rank*. Isto é, ao dividir a classificação de qualquer nó da rede por 256, que é definido como o incremento de classificação de salto mínimo do RPL, tem-se o valor da contagem de saltos.

A informação envolvendo o padrão mínimo de incremento do rank (DEFAULT MIN HOP RANK INCREASE) está presente na documentação do IETF do protocolo (BRANDT; HUI; KELSEY, 2012). Além disso, ele é utilizado em alguns trechos do algoritmo apresentado no Anexo A dentro da seção 5, para a função objetivo OF0. Essa configuração resulta em um número inteiro de 8 bis.

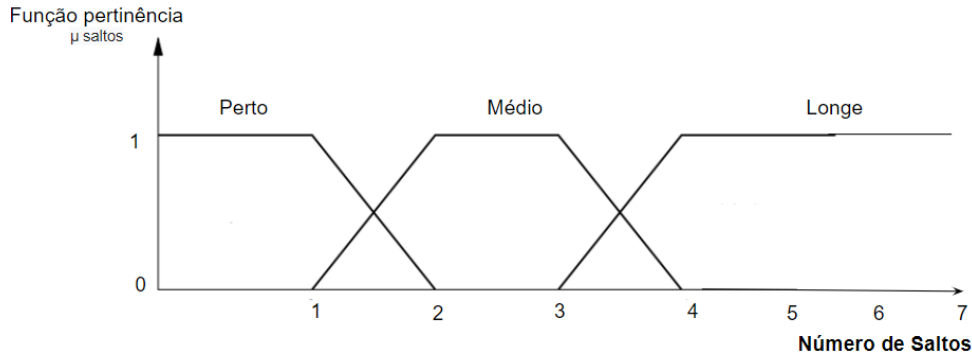
O incremento máximo do rank máximo definido como:

$$MaxRankIncrSalto = 7 * MinRankIncSalto \quad (3.7)$$

A equação 3.7 implica que o rank máximo em uma rede é $7 \times 256 = 1792$. É possível obter a contagem máxima de saltos definida pelo COntiki RPL como 7, valor utilizado neste projeto

Além disso, estes valores são ajustáveis dependendo da necessidade e escolha de design. Como *default*, o nó raiz tem uma contagem de saltos de zero que é necessária ao iniciar na formação do DODAG, transmitindo a mensagem DIO contendo o valor de classificação durante a implantação, como explicado na seção 2.3.

Figura 26: Função Pertinência para Número de Saltos



Fonte: Autor

$$perto(nSaltos) = \begin{cases} 1 & \text{se } nSaltos \leq 1 \\ \frac{nSaltos-2}{1-2} & \text{se } 1 < nSaltos < 2 \\ 0 & \text{se } nSaltos \geq 2 \end{cases} \quad (3.8)$$

$$medio(nSaltos) = \begin{cases} 0 & \text{se } nSaltos \leq 1 \\ \frac{nSaltos-1}{2-1} & \text{se } 1 < nSaltos < 2 \\ 1 & \text{se } 2 < nSaltos < 3 \\ \frac{nSaltos-4}{3-4} & \text{se } 3 < nSaltos < 4 \\ 0 & \text{se } nSaltos \geq 4 \end{cases} \quad (3.9)$$

$$longe(nSaltos) = \begin{cases} 0 & \text{se } nSaltos \leq 3 \\ \frac{nSaltos-3}{4-3} & \text{se } 3 < nSaltos < 4 \\ 1 & \text{se } nSaltos \geq 4 \end{cases} \quad (3.10)$$

3.4.3 Motor de Inferência

No Motor de Inferência Fuzzy são usadas as regras para avaliação do sistema e, assim, propor significados e resultados às diferentes combinações das três entradas sobreditas. Dentro desse módulo, a base das regras de avaliação envolve o *antecedente* das variáveis linguísticas, o operador lógico da decisão, o valor fuzzy definido, o valor antecedente, além das variáveis de entrada e o consequente valor de saída de qualidade.

Nesse sentido, com base em inúmeras análises bibliográficas e simulações envolvendo algumas das métricas hoje presentes dentro da documentação do protocolo RPL aplicado a mobilidade (BRANDT; HUI; KELSEY, 2012), o ambiente de simulação e algoritmo fuzzy são implementados variando os pesos nos parâmetros de entrada e acompanhando o

valor do antecedente da métrica. Sendo M a métrica de entrada do sistema fuzzy, temos na equação 3.11 a relação direta entre os antecedentes com o peso de cada métrica (P entre 0 e 1) e o valor fuzzy definido.

$$\text{Antecedente}(M) = P(M) * \text{ValorFuzzy}(M) \quad (3.11)$$

Ao serem variados, os valores de entrada ativam algumas das regras do conjunto para o valor antecedente. Logo em seguida há a associação relacionada ao valor de qualidade da saída que, no caso, simboliza a qualidade do nós.

Na Tabela 5 são as apresentadas as três regras utilizadas no projeto, variando os três elementos de entrada já definidos e que serão utilizados na obtenção dos dados de qualidade do nó a serem implementados na OF.

Nesse sistema, um total de 27 regras são geradas multiplicando o valor do peso e a pontuação do valor fuzzy. Assim, um valor no intervalo de 1 e 3 é obtido sinalizando o antecedente. Por meio do antecedente, os índices pré-definidos de qualidade são associados. Nesse projeto temos os índices de qualidade do nó como: Excelente, Muito Bom, Bom, Mediano, Ruim, Muito Ruim, Horrível. Tais detalhes serão apresentados na seção seguinte.

Tabela 5: Regras de avaliação utilizadas no processo de Inferência

Regra de Avaliação	Peso da variável de entrada		
	P(Delay)	P(ETX)	P(Nsaltos)
1	0,5	0,3	0,2
2	0,6	0,25	0,15
3	0,3	0,3	0,4

Na Tabela 6 pode-se observar os pesos $P = [0,5 \ 0,3 \ 0,2]$ associados às métricas *Delay*, *ETX* e número de saltos, respectivamente. Como observado temos uma qualidade "Excelente" no cenário no qual *delay* é considerado baixo, *ETX* é pequeno e número de saltos indicado como perto. Esse comportamento se repete em todos os outros cenários por se tratar da melhor situação possível. O mesmo vale para a pior qualidade possível, denominada "Horrível". Nesse ponto temos um *Delay* longo, um *ETX* alto e um número de saltos considerado longe, ambos os valores tenho valor 3.

Tabela 6: Obtendo valor de saída de Qualidade do nó - Regra 1

Delay P = 0,5	Valor Fuzzy Definido	ETX P = 0,3	Valor Fuzzy Definido	Número de Saltos P = 0,2	Valor Fuzzy Definido	Valor Antecedente	Qualidade
Baixo	1	Pequeno	1	Perto	1	1	Excelente
Baixo	1	Pequeno	1	Médio	2	1,2	Muito Bom
Baixo	1	Médio	2	Perto	1	1,3	Muito Bom
Baixo	1	Pequeno	1	Longe	3	1,4	Muito Bom
Baixo	1	Médio	2	Médio	2	1,5	Bom
Médio	2	Pequeno	1	Perto	1	1,5	Bom
Baixo	1	Alto	3	Perto	1	1,6	Bom
Baixo	1	Médio	2	Longe	3	1,7	Bom
Médio	2	Pequeno	1	Médio	2	1,7	Bom
Baixo	1	Alto	3	Médio	2	1,8	Bom
Médio	2	Médio	2	Perto	1	1,8	Mediano
Médio	2	Pequeno	1	Longe	3	1,9	Mediano
Baixo	1	Alto	3	Longe	3	2	Mediano
Médio	2	Médio	2	Médio	2	2	Mediano
Longo	3	Pequeno	1	Perto	1	2	Mediano
Médio	2	Alto	3	Perto	1	2,1	Ruim
Médio	2	Médio	2	Longe	3	2,2	Ruim
Longo	3	Pequeno	1	Médio	2	2,2	Ruim
Médio	2	Alto	3	Médio	2	2,3	Ruim
Longo	3	Médio	2	Perto	1	2,3	Ruim
Longo	3	Pequeno	1	Longe	3	2,4	Ruim
Médio	2	Alto	3	Longe	3	2,5	Muito Ruim
Longo	3	Médio	2	Médio	2	2,5	Muito Ruim
Longo	3	Alto	3	Perto	1	2,6	Muito Ruim
Longo	3	Médio	2	Longe	3	2,7	Muito Ruim
Longo	3	Alto	3	Médio	2	2,8	Horrível
Longo	3	Alto	3	Longe	3	3	Horrível

Um dos detalhes que podem ser notados no andamento das atribuições da regra de inferência tem-se a presença de mais elementos de uma mesma qualidade em um das 27 regras estipuladas. No caso da Tabela 7, temos aqui $P = [0,6 \ 0,25 \ 0,15]$ para as mesmas respectivas métricas citadas. E nesse sentido, de acordo com a quantidade de qualidades num mesmo espectro, é possível notar 3 cenários com qualidade "Horrível", sendo uma atribuição de regra menos tolerante a casos como esse.

Por outro lado, na categoria "Muito Bom" temos 5 combinações possíveis, atingindo valores de antecedentes entre 1,15 e 1,4. No primeiro cenário são notados apenas 3 deles, variando de 1,2 a 1,4.

Tabela 7: Obtendo valor de saída de Qualidade do nó - Regra 2

Delay P = 0,6	Valor Fuzzy Definido	ETX P = 0,25	Valor Fuzzy Definido	Número de Saltos P = 0,15	Valor Fuzzy Definido	Valor Antecedente	Qualidade
Baixo	1	Pequeno	1	Perto	1	1	Excelente
Baixo	1	Pequeno	1	Médio	2	1,15	Muito Bom
Baixo	1	Médio	2	Perto	1	1,2	Muito Bom
Baixo	1	Pequeno	1	Longe	3	1,3	Muito Bom
Baixo	1	Médio	2	Médio	2	1,35	Muito Bom
Baixo	1	Alto	3	Perto	1	1,4	Muito Bom
Baixo	1	Médio	2	Longe	3	1,5	Bom
Baixo	1	Alto	3	Médio	2	1,55	Bom
Médio	2	Pequeno	1	Perto	1	1,65	Bom
Baixo	1	Alto	3	Longe	3	1,7	Bom
Médio	2	Pequeno	1	Médio	2	1,8	Mediano
Médio	2	Médio	2	Perto	1	1,85	Mediano
Médio	2	Pequeno	1	Longe	3	1,95	Mediano
Médio	2	Médio	2	Médio	2	2	Mediano
Médio	2	Alto	3	Perto	1	2,05	Mediano
Médio	2	Médio	2	Longe	3	2,15	Rum
Médio	2	Alto	3	Médio	2	2,2	Ruim
Longo	3	Pequeno	1	Perto	1	2,3	Ruim
Médio	2	Alto	3	Longe	3	2,35	Ruim
Longo	3	Pequeno	1	Médio	2	2,45	Ruim
Longo	3	Médio	2	Perto	1	2,5	Muito Ruim
Longo	3	Pequeno	1	Longe	3	2,6	Muito Ruim
Longo	3	Médio	2	Médio	2	2,65	Muito Ruim
Longo	3	Alto	3	Perto	1	2,7	Muito Ruim
Longo	3	Médio	2	Longe	3	2,8	Horrível
Longo	3	Alto	3	Médio	2	2,85	Horrível
Longo	3	Alto	3	Longe	3	3	Horrível

Por fim, adotando $P = [0,3 \ 0,3 \ 0,4]$ e, dessa forma, aumentando a influência do ETX e do número de saltos nos critérios de decisão, é possível acompanhar os resultados na Tabela 8. Nela é possível notar um aumento das qualidades "Bom" e "Mediano" quando comparado com o critério anterior.

Adotando as variações nos pesos de cada um dos valores médios das métricas, é possível acompanhar as tabelas citadas e a implementação dentro do algoritmo fuzzy completo. Nas seções seguintes será possível avaliar de forma prática quais e de que forma elementos impactam as métricas selecionadas no projeto, adotando o simulador Cooja e o modelo de mobilidade proposto na topologia *Manhattan*.

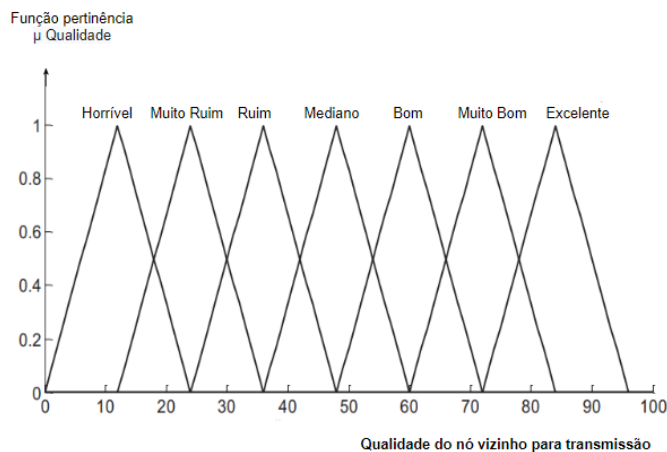
3.4.4 Defuzzyficação

Trata-se do estágio final - o quarto passo no exemplo de ilustração do sistema da Figura 15 -, o qual envolve a realização de uma avaliação completa para obter da saída final exatas com as regras apresentadas. Ou seja, temos um dado exato que concentra a qualidade de um nó a partir de ponderações dos 3 elementos de entrada apresentados.

Tabela 8: Obtendo valor de saída da Qualidade do nó - Regra 3

Delay P = 0,3	Valor Fuzzy Definido	ETX P = 0,3	Valor Fuzzy Definido	Número de Saltos P = 0,4	Valor Fuzzy Definido	Valor Antecedente	Qualidade
Baixo	1	Pequeno	1	Perto	1	1	Excelente
Médio	2	Pequeno	1	Perto	1	1,3	Muito Bom
Baixo	1	Médio	2	Perto	1	1,3	Muito Bom
Baixo	1	Pequeno	1	Médio	2	1,4	Muito Bom
Longo	3	Pequeno	1	Perto	1	1,6	Bom
Médio	2	Médio	2	Perto	1	1,6	Bom
Baixo	1	Alto	3	Perto	1	1,6	Bom
Médio	2	Pequeno	1	Médio	2	1,7	Bom
Baixo	1	Médio	2	Médio	2	1,7	Bom
Baixo	1	Pequeno	1	Longe	3	1,8	Mediano
Longo	3	Médio	2	Perto	1	1,9	Mediano
Médio	2	Alto	3	Perto	1	1,9	Mediano
Longo	3	Pequeno	1	Médio	2	2	Mediano
Médio	2	Médio	2	Médio	2	2	Mediano
Baixo	1	Alto	3	Médio	2	2	Mediano
Médio	2	Pequeno	1	Longe	3	2,1	Ruim
Baixo	1	Médio	2	Longe	3	2,1	Ruim
Longo	3	Alto	3	Perto	1	2,2	Ruim
Longo	3	Médio	2	Médio	2	2,3	Ruim
Médio	2	Alto	3	Médio	2	2,3	Ruim
Longo	3	Pequeno	1	Longe	3	2,4	Ruim
Médio	2	Médio	2	Longe	3	2,4	Muito Ruim
Baixo	1	Alto	3	Longe	3	2,4	Muito Ruim
Longo	3	Alto	3	Médio	2	2,6	Muito Ruim
Longo	3	Médio	2	Longe	3	2,7	Muito Ruim
Médio	2	Alto	3	Longe	3	2,7	Muito Ruim
Longo	3	Alto	3	Longe	3	3	Horrível

Figura 27: Função Pertinência da Qualidade do nó vizinho - Saída do sistema



Fonte: Autor

O centro de gravidade é o método de defuzzificação utilizado no projeto, no qual é aplicado à função de associação para encontrar o centro agregado a regra.

A equação 2.3 descreve o cálculo executado para a saída exata. O valor de qualidade exato (Q) obtido a partir da performance da operação fuzzy em cada pai é então usado para selecionar o pai com a melhor qualidade.

4 RESULTADOS E DISCUSSÃO

Considerando o cenário de mobilidade apresentado e aplicação ao contexto dos VANETs, nessa seção serão abordadas algumas das principais métricas de eficiência do protocolo RPL em ambientes móveis comparando as funções objetivo consolidadas como padrão, OF0 e MHROF, com os modelos propostos com as regras: FuzzyC1, FuzzyC2, FuzzyC3. Respectivamente elas acompanham as regras de ponderação apresentadas na Tabela 5.

Além da análise variando as funções objetivo, outros elementos são variados em algumas das simulações - quando considerados pertinentes. Analisando o PDR (taxa de entrega do pacote) e *Delay* de ponta a ponta, é medido o comportamento das OF's na variação da frequência no envio dos pacotes - 1pc/s (pacote/segundo), 3pc/2s e 2p/s.

Outro elemento variante em cada simulação é o cenário de mobilidade. Como apresentado na seção anterior, temos 20 nós fixos e 5 nós móveis (20% de nós são móveis na rede) movimentando-se a 10m/s (36 km/h) sob área de 500m X 500m. Em simulações seguintes foi analisado o impacto com o aumento na quantidade de nós móveis (NM), alterando para 10 (33% de mobilidade na rede) e 20 (50% de mobilidade) nos cenários seguintes.

Quando analisada a terceira métrica para cenários móveis, o consumo de energia, as medições envolvem agora a variação de velocidade dos nós móveis (mantendo o número de envios de pacote fixo em 1 pc/s), realizando o projeto em 10m/s e 20m/s. Há aqui uma variação na velocidade por ser considerada importante em toda a análise envolvendo carros e mobilidade urbana.

4.1 Avaliação de Performance

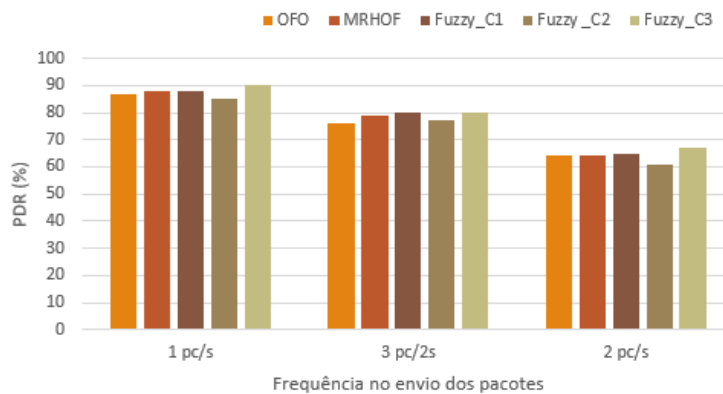
4.1.1 Taxa de Entrega de pacotes

Com o objetivo de apresentar melhor confiabilidade da rede, a métrica PDR (taxa de entrega dos pacotes) é analisada. Definida pela equação 4.1, é a relação direta entre a quantidade de pacotes enviados e recebidos no nó raiz.

Fornecendo informações importantes relativas a perda de pacote dentro do ambiente de mobilidade e considerando as características apresentadas no início da seção: variação no envio do pacote (1pc/s, 3pc/2s e 2pc/s), velocidade média de 10m/s, 3 cenários de mobilidade (25, 30 e 40 nós) e as cinco diferentes funções objetivo em detalhe no projeto.

$$PDR(\%) = \frac{\sum Pacotes\ Recebidos}{\sum Pacotes\ Enviados} * 100 \quad (4.1)$$

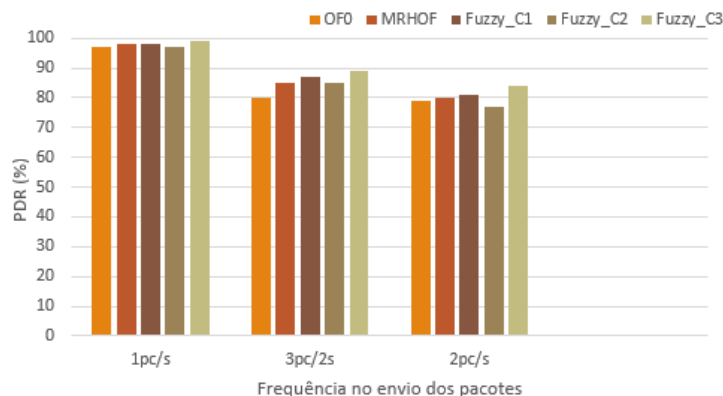
Figura 29: Comparativo da taxa de entrega de pacote com 10 nós móveis



Fonte: Simulador Cooja

Nas Figuras 28, 29 e 30 são ilustrados os resultados das simulados de PDR. Nas simulações apresentadas são considerados 400 pacotes em média e a avaliação da entrega de pacote nos permite identificar o comportamento e influência dos nós móveis nas recepções desses dados pelo nó raiz.

Figura 28: Comparativo da taxa de entrega de pacote com 5 nós móveis



Fonte: Simulador Cooja

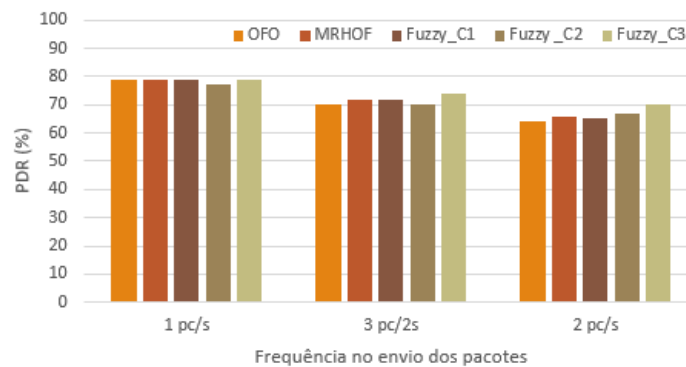
É perceptível que a presença de um número maior de nós móveis causa uma menor taxa de entrega. Esse comportamento ocorre pois nós fixos tem maiores chances de se conectar a um nó que esteja enviando DIOS com um rank menor que o do atual pai do nó fixo em questão, prejudicando a performance do PDR.

Tais índices são "puxados para baixo" pelos nós móveis, que apresentam uma PDR menor que nós fixos e o aumento dessa malha de mobilidade (e também da velocidade dos nós). Ou seja, pode-se notar um teto na PDR variando de 100 para 90 e, por fim, para 80, ao serem adicionados 5 nós móveis a cada nova interação.

Há também a variação na frequência dos pacotes, diminuindo substancialmente as taxas de recepção pois um aumento de dados em trânsito podem causar congestionamento no fluxo de informação e causando dados sendo perdidos.

Como reforçado acima, a partir desses resultados de simulação é possível observar que a perda de pacote é baixa em pequena densidade de nós, tornando-se grave sob uma alta densidade de objetos móveis.

Figura 30: Comparativo da taxa de entrega de pacote com 20 nós móveis



Fonte: Simulador Cooja

Sob termos práticos, todos os cenários envolvendo as 5 funções objetivo acompanham um comportamento global quando são atingidas por aumento na frequência (todas tem PDR caindo nesse cenário) e também quando apresentam nós móveis (PDR decrescente com aumento de nós móveis).

As implementações Fuzzy apresentam uma taxa de entrega maior do que com OF0 e resultados semelhantes que uma rede baseada em ETX (MRHOF). Destaque para o cenário 3, apresentando um $P = [0,3 \ 0,4 \ 0,3]$, equilibrando os critérios de entrada e sobressaindo, mesmo que pouco a métrica ETX, valorizando significativamente o *delay* e o número de saltos.

A razão da alta perda de pacotes na rede baseada em OF0 é que o mesmo não promove links de alta qualidade, e ao escolher o pai com classificação mínima, esse pai pode estar congestionado e assim irá descartar os pacotes de dados, o que é intensificado com o aumento de nós e da taxa de envio.

Como visto na Tabela 9, a rede de OF MRHOF, baseada em ETX, possui desempenho similar aos de implementação fuzzy, como FuzzyC2 e isso se dá devido ao uso do mesmo estimador de qualidade de link que é ETX. Minimizar o ETX ao selecionar o próximo salto implica um caminho com baixa taxa de perda de pacotes.

Um diferencial da implementação quando comparado ao MRHOF, mesmo que superficial, ocorre pois os primeiros executam a operação da transmissão com número menor de pacotes de dados, minimizando colisões e congestionamento da rede, que ocorrem, mesmo que em pequena escala, no MRHOF. Esse fato é de extrema importância quando lidamos com grande número de nós e consequente tráfego de dados.

Tabela 9: Levantamento dos valores de PDR obtidos nos cenários propostos

Cenário	OF	PDR (%)		
		1pc/s	3pc/2s	2pc/s
25 nós (5 móveis)	OF0	97	90	79
	MRHOF	98	85	80
	Fuzzy_C1	98	87	81
	Fuzzy_C2	97	85	77
	Fuzzy_C3	99	89	84
30 nós (10 móveis)	OF0	87	76	64
	MRHOF	88	79	64
	Fuzzy_C1	88	80	65
	Fuzzy_C2	85	77	61
	Fuzzy_C3	90	80	67
40 nós (20 móveis)	OF0	79	70	64
	MRHOF	79	72	66
	Fuzzy_C1	79	72	65
	Fuzzy_C2	77	70	67
	Fuzzy_C3	79	74	70

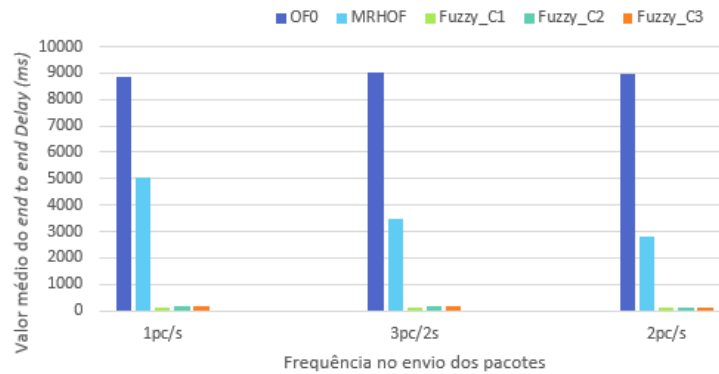
4.1.2 Delay End-To-End

Analisando o comportamento da métrica de *Delay end-to-end*, de suma importância em sistemas móveis e que carecem de informação em tempo real, podem ser notadas melhorias muito significativas em todos os cenários de frequência no pacote e também ao variarmos a quantidade de nó móveis.

Sendo uma métrica que, quando não satisfatória, pode causar loops na rede e perdas de pacotes, a melhoria no aspecto envolvendo o atraso de ponta a ponta é um ganho importante no projeto.

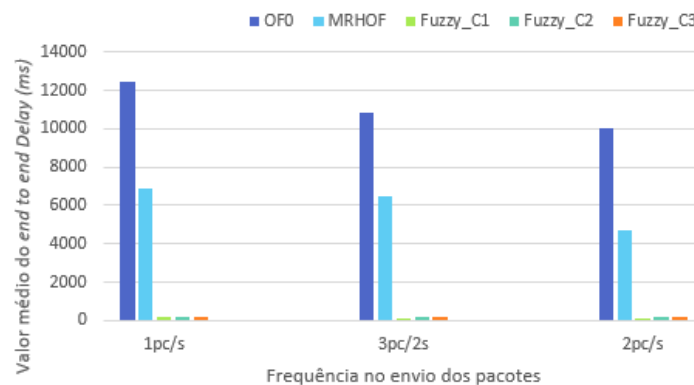
As Figuras 31, 32 e 33 mostram a variação do atraso de ponta a ponta (em ms) para FuzzyC1, FuzzyC2, FuzzyC3, MRHOF e OF0 dentro do tempo de simulação. Pode-se observar que o atraso de ponta a ponta com as implementações da OF-FuzzyCx estão sempre abaixo das MRHOF e OF0, exibindo uma fragilidade - já esperada - das duas funções objetivo quando imersos em um ambiente de mobilidade.

Figura 31: Comparativo do *delay* na entrega do pacote com 5 nós móveis



Fonte: Simulador Cooja

Figura 32: Comparativo do *delay* na entrega do pacote com 10 nós móveis



Fonte: Simulador Cooja

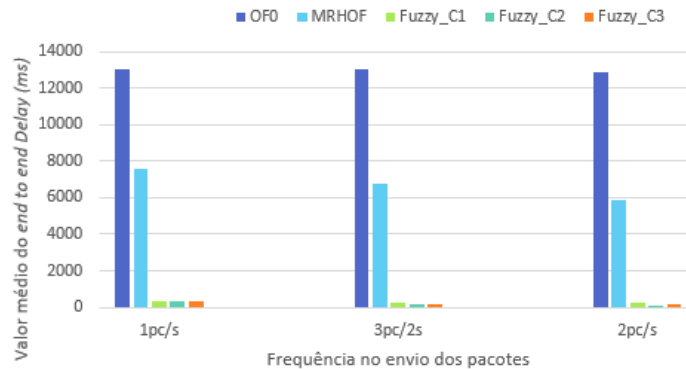
Mesmo que elementos presentes (e com forte impacto) nessas OF's, como o número de saltos, favoreçam caminhos com menos saltos, esses caminhos são geralmente mais longos e com conectividade fracas.

Por outro lado, o ETX (muito presente na função MRHOF) também não está ciente dos atrasos devidos à interferência nos links ou filas de sobrecarga nas mensagens de controle (DIS, DIO) dos nós quando estes são transmitidos; assim sendo, enviar um pacote com menos retransmissões não significa enviando-o em um link mais rápido.

O resultado apresentado deve-se principalmente à média contagem de saltos que é minimizada e o atraso de ponta a ponta que é considerado ao decidir sobre o próximo salto, variando seu peso no espectro fuzzy de 0,6 a 0,3.

Estes resultados são considerados promissores, já que permite minimizar o atraso em relação às funções objectivo existentes. Quando observa-se especificamente a função OF0, por exemplo, é visto uma maior indução na latência quando comparado com os outros

Figura 33: Comparativo do *delay* na entrega do pacote com 20 nós móveis



Fonte: Simulador Cooja

quatro esquemas, embora permita minimizar a contagem de saltos. Portando caminhos confestionados são muito impactantes no processo.

Além disso, uma maior taxa de perda de pacote presente nas duas funções padrão - principalmente na OF0 - também impacta no atraso, prejudicando a performance.

Por fim, como esperado, um aumento na taxa de envio dos pacotes reduz o tempo de atraso entre os pontos e o aumento no número de nós da rede - além de serem nós móveis - gera um aumento no tempo de *delay*. Isso ocasionado por nós que saem do caminho da DODAG, além de existirem mais mensagens e, por conseguinte, uma sobrecarga no fluxo desses dados de controle, ocasionando um tempo maior para transmissão completa do dado.

Os dados completos comparativos entre as funções objetivo podem ser encontradas na Tabela 10. Analisando os dados somente dentro do universo fuzzy, é possível validar que há um melhor comportamento dos FuzzyC1 em ambientes com 5 a 10 nós móveis (25 a 30 nós na rede). Essa priorização do fator de entrada de atraso nos dois primeiros modelos ($P = 0,5$ e $0,6$, respectivamente) pode ser um determinante nesse ambiente, tendo um limitante no número de nós que, quando crescente, estabilizam a operação devido os limites de abrangência do raio da árvore de roteamento e o aumento no número de mensagens de controle, como comentado acima. Mesmo diante de um cenário de estabilidade há uma melhoria muito significativa.

Tabela 10: Levantamento dos valores de *delay* obtidos nos cenários propostos

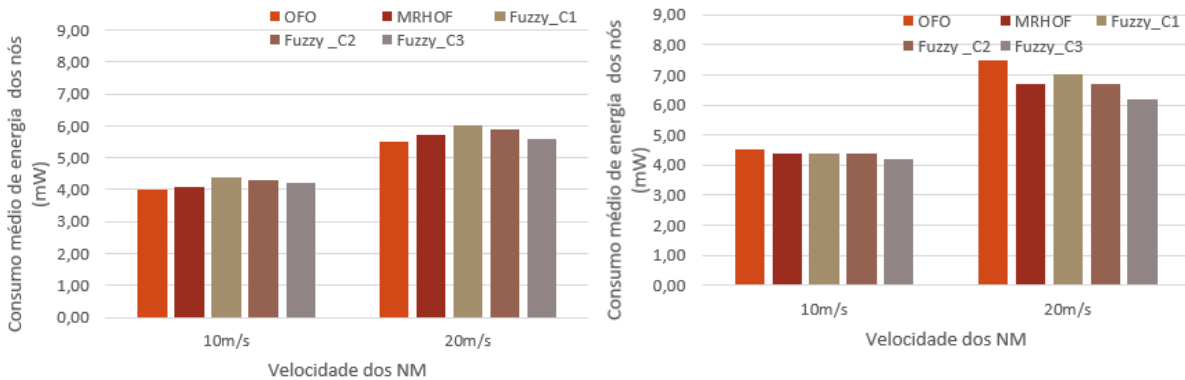
Cenário	OF	Delay (ms)		
		1pc/s	3pc/2s	2pc/s
25 nós (5 móveis)	OF0	8870	9001	8978
	MRHOF	5058	3450	2801
	Fuzzy_C1	124	129	110
	Fuzzy_C2	167	156	124
	Fuzzy_C3	170	144	130
30 nós (10 móveis)	OF0	12470	10875	10032
	MRHOF	6905	6450	4671
	Fuzzy_C1	178	144	141
	Fuzzy_C2	197	204	189
	Fuzzy_C3	199	217	201
40 nós (20 móveis)	OF0	13056	13001	12897
	MRHOF	7600	6781	5866
	Fuzzy_C1	294	287	284
	Fuzzy_C2	309	256	249
	Fuzzy_C3	331	300	291

4.1.2.1 Consumo de Energia e a influência da velocidade

A eficiência energética pode ser medida por meio do total de energia consumida, ou a vida útil da rede, ou mesmo a energia remanescente, além de outros elementos possíveis para essa medição. Executar o projeto com essas propriedades pode ser desafiador, pois elas podem ser contrárias em algum momento. Uma rede mais confiável energeticamente pode ser a que tenha maior número de saltos ou mesmo que tenha maior atraso ponta a ponta.

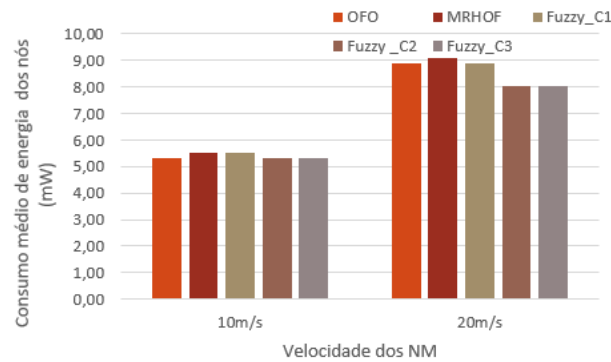
Nesse sentido, são analisados 3 cenários de mobilidade considerando as mesmas topologias de nós apresentadas anteriormente, agora com frequência de envio padronizado em 1pc/s e com variação na velocidade dos nós móveis (NM) analisando em 10 m/s (36km/h) e 20 m/s (72 km/h).

Figura 34: Consumo médio de energia nos nós. Os 3 cenários propostos e a influência da velocidade dos nós móveis.



(a) Cenário com 5 nós móveis

(b) Cenário com 10 nós móveis



(c) Cenário com 20 nós móveis

Fonte: Simulador Cooja

Os resultados apresentados na Figura 34 validam que o aumento da velocidade atinge diretamente no gasto energético médio da rede, ocasionado pois o aumento da velocidade exige um número maior de mensagens para a entrada na rede.

Mesmo não utilizando diretamente essa métrica na execução da implementação em Fuzzy, a rede com as OF's FuzzyC2 e FuzzyC3 exibiram um consumo menor quando apresentados dentro do cenário 3 com 40 nós na rede. Nesse momento temos uma melhoria no gasto, não observado nos cenários anteriores.

Tabela 11: Levantamento dos valores de consumo médio de energia nos nós obtidos nos cenários propostos

Cenário	OF	Energia média consumida (mW)	
		10m/s	20m/s
25 nós (5 móveis)	OF0	4,01	5,50
	MRHOF	4,10	5,70
	Fuzzy_C1	4,40	6,00
	Fuzzy_C2	4,29	5,90
	Fuzzy_C3	4,20	5,60
30 nós (10 móveis)	OFO	4,54	7,50
	MRHOF	4,39	6,70
	Fuzzy_C1	4,40	7,00
	Fuzzy_C2	4,38	6,70
	Fuzzy_C3	4,21	6,20
40 nós (20 móveis)	OFO	5,30	8,90
	MRHOF	5,49	9,10
	Fuzzy_C1	5,50	8,90
	Fuzzy_C2	5,29	8,00
	Fuzzy_C3	5,32	8,00

5 CONCLUSÃO

A eficiência e confiabilidade das LLN's na presença de mobilidade são consideradas desafiadoras devido à natureza dos nós e à inerente dinâmica criada pelo nó móvel, possibilitando que informações não cheguem ao seu destino devido a saída do nó da DODAG, por exemplo. Este trabalho de implementação e simulação foi capaz de melhorar o resultado de alguns desses problemas, estudando o protocolo de roteamento RPL disponível no sistema operacional Contiki.

Após a validação do protocolo no simulador Cooja, algumas das métricas relevantes do nó e da rede foram utilizadas para gerenciar os problemas de mobilidade e garantindo um roteamento de dados utilizando um modelo de sistema de controle fuzzy. Isso foi possível com o uso de três métricas de roteamento: *delay end-to-end*, ETX e a contagem de saltos. Esses parâmetros foram selecionados devido à sua importância em redes móveis, impactando, por exemplo, diretamente em comunicações que exigem informações em tempo real. Os três parâmetros serviram como entrada para o modelo de lógica fuzzy, ponderando seus pesos e medindo a qualidade do nó vizinho a cada interação no algoritmo.

Sucessivas simulações foram realizadas usando algumas métricas de avaliação para testar vários cenários de ponderação na mobilidade: *delay end-to-end*, PDR (taxa de entrega de pacote) e energia consumida nos nós. As ponderações foram utilizadas no ajuste do sistema para identificar o melhor cenário em cada um dos testes, garantindo uma entrega confiável de dados. Como ambiente de simulação, foi utilizado um modelo de mobilidade Manhattan aplicado a VANETs, mantendo a velocidade dos carros em 10 e 20m/s movimentando-se de forma linear sobre o cenário.

Dentre os resultados podem ser destacados a amenização na perda de pacotes em ambientes móveis quando consideramos as implementações, principalmente o modelo FuzzyC3, o qual apresenta uma ponderação próxima a uma divisão exata entre as três métricas de entrada ($P(\text{delay}) = 0,3$; $P(\text{ETX}) = 0,4$ e $P(\text{nSaltos}) = 0,3$) e consequentemente, notou-se uma taxa de entrega maior.

Uma métrica com resultado surpreendente está associada ao atraso de ponta a ponta. Implementações considerando a qualidade do nó vizinho como estas apresentadas minimizam o tempo para índices para valores entre 2% e 3% do valor total quando comparados com OF0 e entre 2,5% e 4% quando comparados com o MRHOF. Isso ocorre pois as funções padronizadas ora não consideram a sobrecarga no caminho escolhido, ora desconsideram a conectividade fraca entre nós.

Por fim, uma análise envolvendo gasto energético e variação da velocidade apresenta resultados promissores, principalmente quando ocorre a comparação dentro de um cenário

com 50% dos nós móveis. As implementações fuzzy exercem um menor variação no consumo médio de energia por nó, o que afeta a longo prazo o tempo de vida desses sensores. A variação de velocidade foi explicitada nessa etapa, pois foi notada uma influência significativa em outras simulações fora do projeto.

Nos três principais experimentos, foi possível validar a importância de uma implementação para o uso do protocolo RPL em ambientes móveis e, com o intuito de refinar e tornar os resultados mais complexos no futuro, uma análise da influência geográfica de ambientes VANET's, novos modelos de mobilidade ou mesmo ponderações considerando outros elementos de entrada no motor de inferência, como o indicador de potência do sinal recebido (RSSI, do inglês *Received Signal Strength Indicator*) por exemplo, podem ser bons caminhos para tornar o projeto com novos resultados e dimensões.

REFERÊNCIAS

AFONSO, B.; PEREIRA, R. B.; PEREIRA, M. F. Utilização da internet das coisas para o desenvolvimento de miniestação de baixo custo para monitoramento de condições do tempo em áreas agrícolas. Anais da Escola Regional de Informática da Sociedade Brasileira de Computação (SBC) – Regional de Mato Grosso, p. 183–189, Nov 2015. Disponível em: <<http://anaiserimt.ic.ufmt.br/index.php/erimt/article/view/50>>.

ALAM, M.; SHER, M.; HUSAIN, S. A. Integrated mobility model (imm) for vanets simulation and its impact. International Conference on Emerging Technologies, p. 29–38, Nov 2009. Disponível em: <https://www.researchgate.net/publication/224089964_Integrated_Mobility_Model_IMM_for_VANETs_simulation_and_its_impact>.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. Computer Networks, p. 19, sep 2012. Disponível em: <<https://www.cs.mun.ca/courses/cs6910/IoT-Survey-Atzori-2010.pdf>>.

BEDI, P.; JINDAL, V. Use of big data technology in vehicular ad-hoc networks. International Conference on Advances in Computing, Communications and Informatics (ICACCI-2014), p. 29–38, Sep 2014. Disponível em: <https://www.researchgate.net/publication/266148960_Use_of_Big_Data_Technology_in_Vehicular_Ad-hoc_Networks>.

BIDGOLI, H. **The Internet Encyclopedia**. First edition. [S.l.]: John Wiley Sons, 2004.

BLOCK, I. **Invisible Landscapes installations explore how virtual reality will change architecture**). 2019. <<https://www.dezeen.com/2019/04/04/virtual-reality-installations-invisible-landscapes-royal-academy/>>. Accessed: 2019-20-04.

BRACHMAN, A. Rpl objective function impact on llns topology and performance. Springer, p. 340–350, Oct 2013. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-40316-3_30>.

BRANDT, A.; HUI, J.; KELSEY, R. Rpl: Ipv6 routing protocol for low-power and lossy networks. Internet Engineering Task Force, p. 154, 2012. Disponível em: <<https://tools.ietf.org/html/rfc6550>>.

BRITO, A. C. **O que o futuro reserva para a Internet das Coisas (IoT)?** . 2019. <<https://itforum365.com.br/o-que-o-futuro-reserva-para-a-internet-das-coisas-iot/>>. Accessed: 2019-29-03.

CAVALCANTI, J. H. F.; MELO, H. de; SOUTO, C. da R.; CAVALCANTI, M. T. **LÓGICA FUZZY APLICADA ÀS ENGENHARIAS**. First edition. [S.l.: s.n.], 2012.

CAYIRCI, E.; COPLU, T.; MIROGLU, O. Power aware many to many routing in wireless sensor and actuator networks. Proc. of the Second European Workshop on Wireless Sensor Networks, p. 9, Feb 2005. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/1462016/>>.

CHARARA, S. **What are the IoT challenges holding back adoption?** . 2018. <<https://www.raconteur.net/technology/iot-challenges>>. Accessed: 2019-01-05.

CONTIKI-OS. **Contiki: The Open Source OS for the Internet of Things**. 2013. <<http://www.contiki-os.org/index.html>>. Accessed: 2019-08-05.

COUTO, D. S. J. D.; AGUAYO, D.; BICKET J. AND MORRIS, R. A high-throughput path metric for multi-hop wireless routing. M.I.T. Computer Science and Artificial Intelligence Laboratory, p. 1–13, 2003. Disponível em: <<https://pdos.csail.mit.edu/papers/grid:mobicom03/paper.pdf>>.

CROCETTA, T. B.; OLIVEIRA, S. R. de; LIZ, C. M. de; ANDRADE, A. Virtual and augmented reality technologies in human performance: a review. *Fisioter. mov*, p. 11–12, Oct 2015. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-51502015000400823>.

DEERING, S.; HINDEN, R.; NOKIA. Internet protocol, version 6 (ipv6) specification. Network Working Group, p. 39, Dec 1998. Disponível em: <<https://tools.ietf.org/html/rfc2460>>.

DESHPANDE, P. **Mobility of Nodes in Cooja**. 2015. <https://anrg.usc.edu/contiki/index.php/Mobility_of_Nodes_in_Cooja>. Accessed: 2019-01-04.

DOMINGUE, J.; DOMINGUE, J.; GALIS, A.; GALIS, A.; GAVRAS, A. The future internet – future internet assembly. *Achievements and Technological Promises*, p. 11–12, Nov 2011. Disponível em: <<https://www.springer.com/gp/book/9783642208973>>.

GADDOUR, O.; KOUBÂA, A. Rpl in a nutshell: A survey. *Computer Networks*, p. 16, jun 2012. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S1389128612002423>>.

GADDOUR, O.; KOUBÂA, A.; ABID, M. Quality-of-service aware routing for static and mobile ipv6-based low-power and lossy sensor networks using rpl. *Ad Hoc Networks*, p. 233–256, Oct 2015. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S1570870515000992>>.

GADDOUR, O.; KOUBÂA, A.; CHAUDHRY, S.; TEZEGHDANTI, M. Simulation and performance evaluation of dag construction with rpl. *rd International Conference on Communications and Networking (ComNet)*, Jun 2012. Disponível em: <<https://ieeexplore.ieee.org/document/6217747>>.

GAI, S. **Internetworking IPv6 with Cisco Routers (McGraw-Hill Computer Communications Series)**. First edition. [S.l.: s.n.], 1997.

GNAWALI, O.; LEVIS, P. The minimum rank with hysteresis objective function. Internet Engineering Task Force (IETF), p. 1–13, Mar 2012. Disponível em: <<https://tools.ietf.org/html/rfc6719>>.

HARROP, P. Wireless sensor networks and the new internet of things. *Energy Harvesting Journal*, p. 5, sep 2012. Disponível em: <<http://www.energyharvestingjournal.com>>.

HARROP, P.; DAS, R. Wireless sensor networks 2012-2022. *IDTechEx*, p. 4, sep 2012. Disponível em: <<https://www.idtechex.com/research/articles/wireless-sensor-networks-and-the-new-internet-of-things-00004770.asp>>.

IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$45 Billion in 2019, Led by the Manufacturing, Consumer, Transportation, and Utilities Sectors.

INCORPORATED, T. I. Msp430f15x, msp430f16x, msp430f161x mixed signal microcontroller. TEXAS INSTRUMENTS INCORPORATED, Out 2002. Disponível em: <<http://www.ti.com/lit/ds/symlink/msp430f1611.pdf>>.

_____. SmartRF transceiver evaluation board “trxeb”: User’s guide. TEXAS INSTRUMENTS INCORPORATED, Out 2017. Disponível em: <<http://www.ti.com/lit/ug/swru294a/swru294a.pdf>>.

J., H.; P, T. Compression format for ipv6 datagrams over ieee 802.15.4-based networks. Internet Engineering Task Force, p. 4, May 2011. Disponível em: <<http://www.rfc-editor.org/info/rfc6282>>.

J, V.; M, K.; K, P.; N, D.; D, B. Routing metrics used for path calculation in low-power and lossy networks. RFC6551, 2013. Disponível em: <<https://tools.ietf.org/html/rfc6551>>.

KEMP, S. **DIGITAL 2019: GLOBAL INTERNET USE ACCELERATES**. 2019. <<https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates>>. Accessed: 2019-10-05.

KOSMATOS, E. A.; TSELIKAS, N. D.; BOUCOUVALAS, A. C. Integrating rfid and smart objects into a unified internet of things architecture. Advances in Internet of Things: Scientific Research, p. 5–12, apr 2011. Disponível em: <<https://www.scirp.org/journal/PaperInformation.aspx?PaperID=4696>>.

LAMAAZI, H.; BENAMAR, N.; IMADUDDIN, M. I.; JARA, A. J. Performance assesment of the routing protocol for low power and lossy networks. International Workshop WINCOM (Wireless Networks and mobile COMMunications), Oct 2015. Disponível em: <<https://ieeexplore.ieee.org/document/7381329>>.

LAMAAZI, H.; BENAMAR, N.; J., A. Study of the impact of designed objective function on the rpl-based routing protocol. IEEE, p. 13, Nov 2017. Disponível em: <https://www.researchgate.net/publication/309693884_Study_of_the_Impact_of_Designed_Objective_Function_on_the_RPL-Based_Routing_Protocol>.

L.A.ZADEH. Fuzzy sets. Information and control, p. 338–353, 1965. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S001999586590241X>>.

LEMOS, A. **A comunicação das Coisas. Teoria Ator-Rede e Cibercultura**. First edition. [S.l.]: Annablume, 2013.

LI, F.; WANG, Y. Routing in vehicular ad hoc networks: A survey. IEEE, p. 12–22, Jun 2017. Disponível em: <<https://ieeexplore.ieee.org/document/4450627>>.

MADAKAM, S.; SIDDHARTH, R.; TRIPATHI, S. Internet of things (iot): A literature review. Journal of Computer and Communications, Mar 2015. Disponível em: <https://www.researchgate.net/publication/280527542InternetofThingsIoTAliterature_Review>.

MITAL, A. **What 2019 Holds for IoT: Just a Happy New Year or More?** . 2018. <<https://blog.sasken.com/what-2019-holds-for-iot-just-a-happy-new-year-or-more>>. Accessed: 2019-11-05.

MOBILE, T. **Augmented Reality in Education** . 2018. <<https://thinkmobiles.com/blog/augmented-reality-education/>>. Accessed: 2019-11-05.

OLSSON, J. "6lowpan demystified". Texas Instrument, p. 4, 2014. Disponível em: <<http://www.ti.com/lit/wp/swry013/swry013.pdf>>.

PHYSORG. **First practical nanogenerator produces electricity with pinch of the fingers**. 2019. <<https://phys.org/news/2011-03-nanogenerator-electricity-fingers.html>>. Accessed: 2019-11-05 - Trecho exibido durante conferência "*the American Chemical Society*".

PICKENS, A.; AIELLO, C. **Customer Success for IoT: The 6 Layers of an Ironclad Strategy**. 2018. <<https://www.gainsight.com/blog/customer-success-iot-6-layers-ironclad-strategy/>>. Accessed: 2019-01-04.

PINTO, P.; PINTO, A.; RICARDO, M. End-to-end delay estimation using rpl metrics in wsn. 2013 IFIP Wireless Days (WD), p. 1–6, Nov 2013. Disponível em: <<https://repositorio.inesctec.pt/bitstream/123456789/4433/1/P-009-4SD.pdf>>.

PISTER, K.; THUBERT, P.; DWARS, S.; PHINNEY, T. Industrial routing requirements in low-power and lossy networks. Network Working Group, p. 27, Oct 2009. Disponível em: <<https://www.rfc-editor.org/rfc/pdf/rfc5673.txt.pdf>>.

RAWAT, P.; SINGH, K. D.; CHAOUCHI, H.; BONNIN, J. M. Wireless sensor networks: recent developments and potential synergies. The Journal of Supercomputing 68, p. 50, mar 2013. Disponível em: <https://www.researchgate.net/publication/258165429_Wireless_sensor_networks_A_survey_on_recent_developments_and_potential_synergies>.

REHMAN, S. ur; KHAN, M. A.; ZIA, T. A.; ZHENG, L. Vehicular ad-hoc networks (vanets) - an overview and challenges. Journal of Wireless Networking and Communications, p. 29–38, Out 2014. Disponível em: <https://www.researchgate.net/publication/258248007_Vehicular_ad-Hoc_networks_VANETs-An_overview_and_challenges>.

ROUSE, M. **WSAN (wireless sensor and actuator network)**. 2015. <<https://internetofthingsagenda.techtarget.com/definition/WSAN-wireless-sensor-and-actuator-network>>. Accessed: 2019-23-04.

SABRI, N.; ALJUNID, S. A.; AHMAD, R. B.; M.F.MALIK; YAHYA, A.; KAMARUDDIN, R.; M.S.SALIM. Towards smart wireless sensor actor networks: Design factors and application. School of Computer and Communication Engineering, p. 5, Sep 2011. Disponível em: <<https://ieeexplore.ieee.org/document/6108808>>.

SANTOS, B. P.; SILVA, L.; CELES, C.; BORGES, J. B.; NETO, B. S. P.; VIEIRA, M. A. M.; VIEIRA, L. F. M.; GOUSSEVSKAIA, O. N.; LOUREIRO, A. Internet das coisas: da teoriaa prática. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuidos, Jun 2016. Disponível em: <<https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>>.

SANTUCCI, G. From internet of data to internet of things. Paper for the International Conference on Future Trends of the Internet, p. 18, Jan 2009. Disponível em: <<https://vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf>>.

SCHRICKTE, L. F.; MONTEZ, C.; OLIVEIRA, R. D.; PINTO, A. R. Integration of wireless sensor networks to the internet of things using a 6lowpan gateway. IEEE, p. 11, 2013. Disponível em: <<https://ieeexplore.ieee.org/document/6825353>>.

SHANG, W.; YU, Y.; DROMS, R.; ZHANG, L. Challenges in iot networking via tcp/ip architecture. NDN Technical Report, p. 4–7, Feb 2016. Disponível em: <<https://named-data.net/wp-content/uploads/2016/02/ndn-0038-1-challenges-iot.pdf>>.

SHELBY, Z.; BORMANN, C. **6LoWPAN The Wireless Embedded Internet**. first edition. [S.l.]: A John Wiley and Sons, 2009.

SHELBY, Z.; HARTKE, K.; BORMANN, C. The constrained application protocol (coap). Internet Engineering Task Force, p. 154, 2014. Disponível em: <<https://tools.ietf.org/html/rfc7252>>.

SHWE, Y. W.; LIANG, Y. C. Smart dust sensor network with piezoelectric energy harvesting. Network Working Group, p. 253–261, Oct 2009. Disponível em: <https://www.researchgate.net/publication/220553400_Smart_dust_sensor_network_with_piezoelectric_energy_harvesting>.

SUTARIA, R.; GOVINDACHARI, R. Understanding the internet of things. Electronic Design, p. 4, May 2013. Disponível em: <<https://www.electronicdesign.com/iot/understanding-internet-things>>.

TANENBAUM, A. **Computer Networks**. 5 th. ed. [S.l.]: Person, 2010.

THOMPCKINS, S. **3 Important Challenges Facing Today's IoT Products**. 2018. <<https://www.centercode.com/blog/2018/01/3-important-challenges-facing-iot-products>>. Accessed: 2019-08-05.

THOMSON, C.; WADHAJ, I.; ROMDHANI, I.; AL-DUBAI, A. Performance evaluation of rpl metrics in environments with strained transmission ranges. IEEE, p. 9, Jun 2017. Disponível em: <<https://ieeexplore.ieee.org/document/7945687>>.

THUBERT, P.; SYSTEMS, C. Objective function zero for the routing protocol for low-power and lossy networks (rpl). Internet Engineering Task Force (IETF), p. 1–13, Mar 2012. Disponível em: <<https://tools.ietf.org/html/rfc6552>>.

T.MELODIA; D.POMPILI; GUNGOR, V.; AKYILDIZ, I. F. Communication and coordination in wireless sensor and actor networks. IEEE Transactions on Mobile Computing, p. 6, Sep 2007. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/4294894/>>.

VERMESAN, O.; FRIESS, D. P.; GUILLEMIN, P.; GUSMEROLI, S.; SUNDMAEKER, H.; BASSI, D. A.; JUBERT, I. S.; MAZURA, D. M.; HARRISON9, D. M.; EISENHAUER, D. M.; DOODY, P. Internet of things strategic research roadmap. The Cluster of European Research Projects, p. 44, sep 2011. Disponível em: <http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf>.

WANG, X.; DING, L.; BI, D.; WANG, S. Energy-efficient optimization of reorganization-enabled wireless sensor networks. sensors. Sensors, p. 7, Sep 2007. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3841847/>>.

WEISER, M. The computer for the 21st century. Scientific American Ubi-comp Paper, 1991. Disponível em: <<https://www.ics.uci.edu/~corps/phaseii/Weiser-Computer21stCentury-SciAm.pdf>>.

YUNIS, J. P.; DUJOVNE, D. Energy efficient routing performance evaluation for llns using combined metrics. IEEE Biennial Congress of Argentina (ARGENCON), p. 1–10, Jun 2014. Disponível em: <<https://ieeexplore.ieee.org/document/6868581/>>.

ČOLAKOVIĆ, A.; HADZIALIC, M. Internet of things (iot): A review of enabling technologies, challenges, and open research issues. Computer Networks 144, p. 23, Jun 2018. Disponível em: <https://www.researchgate.net/publication/258248007_Vehicular_ad-Hoc_networks_VANETs-An_overview_and_challenges>.

Apêndices

Apêndice 1 - Trecho da movimentação do nó 21 a 10 m/s implementado no diretório position.dat com módulo gerador de mobilidade do Cooja

Node	time (s)	x	y
21	1	0	10
21	2	0	20
21	3	0	30
21	4	0	50
21	5	10	50
21	6	10	60
21	7	10	70
21	8	10	80
21	9	10	90
21	10	10	100
21	11	10	110
21	12	10	120
21	13	20	120
21	14	20	110
21	15	20	100
21	16	20	90
21	17	20	80
21	18	20	70
21	19	20	60
21	20	20	50
21	21	20	40
21	22	20	30
21	23	20	20
21	24	20	10
21	25	30	10
21	26	30	30
21	27	30	40
21	28	30	50
21	29	30	60
21	30	30	70
21	31	30	80
21	32	30	90

Apêndice 2 - Segmento do algoritmo do Sistema Inferência Fuzzy implementado com as três entradas definidas e aplicado na biblioteca RPL Contiki

Algoritmo 1: Transmissao_dado

```

1 Inicio
2 se recebeu pacote DIO;
3     Resetar TConn;
4 if Fuzzy_saida < TLimiar; entao
5     va para fase de execucao;
6 else
7     continue Transmissao_dado;
8     end
9 sen o se TMD expirou entao
10    resetar TMD;
11    Envios das mensagens DIS;
12    va para o inicio;
13    senao se TConn expirou entao
14    va para a fase de Execucao;
15    end
16 end

```

Algoritmo 2: Execucao

```

3 Inicio
4     Se recebeu transmissao unicast DIS mensagem; entao
6         guarda counter value C do ultimo pacote DIS;
7         reseta TR com (ws - C) TDIS;
8         se TR expirou ent o
9             calculate average RSSI;
10            dom_pequeno(delay_average);
11            dom_m dio(delay_average);
12            dom_longo(delay_average);
13            dom_pequeno(dio.etx);
14            dom_m dio(dio.etx);
15            dom_largo(dio.etx);
16            dom_perto(diorank/256);
17            dom_m dio(dio.rank/256);
18            dom_longo(dio.rank/256);
19            Fuzzificacao();
20            Regras de Avaliacao();
21            Defuzzificacao();
22            retorna fuzzy_vizinho_n_qualidade_saida;
23            envia transmissao unicast da mensagem DIO message com fuzzy_vizinho_node_
24            senao
25                continue execu o;
26            fim
27        senao
28            continue the Data TX Phase;
29        end
30 end
31 end

```

Anexos

Anexo A - Código para execução da Função Objetivo Of0

```

static void
reset(rpl_dag_t *dag)
{
    PRINTF("RPL: Resetting OF0\n");
}

static rpl_rank_t
calculate_rank(rpl_parent_t *p, rpl_rank_t base_rank)
{
    rpl_rank_t increment;
    if(base_rank == 0) {
        if(p == NULL) {
            return INFINITE_RANK;
        }
        base_rank = p->rank;
    }

    increment = p != NULL ? p->dag->min_hoprankinc : DEFAULT_RANK_INCREMENT;

    if((rpl_rank_t)(base_rank + increment) < base_rank) {
        PRINTF("RPL: OF0 rank %d incremented to infinite rank due to wrapping\n",
            base_rank);
        return INFINITE_RANK;
    }
    return base_rank + increment;
}

static rpl_parent_t *
best_parent(rpl_parent_t *p1, rpl_parent_t *p2)
{
    rpl_rank_t r1, r2;
    rpl_dag_t *dag;

    PRINTF("RPL: Comparing parent ");
    PRINT6ADDR(&p1->addr);
    PRINTF(" (confidence %d, rank %d) with parent ",
        p1->link_metric, p1->rank);
    PRINT6ADDR(&p2->addr);
    PRINTF(" (confidence %d, rank %d)\n",
        p2->link_metric, p2->rank);

    r1 = DAG_RANK(p1->rank, (rpl_dag_t *)p1->dag) * NEIGHBOR_INFO_ETX_DIVISOR +
        p1->link_metric;
    r2 = DAG_RANK(p2->rank, (rpl_dag_t *)p1->dag) * NEIGHBOR_INFO_ETX_DIVISOR +
        p2->link_metric;
    /* Compare two parents by looking both and their rank and at the ETX
       for that parent. We choose the parent that has the most
       favourable combination. */

    dag = (rpl_dag_t *)p1->dag; /* Both parents must be in the same DAG. */
    if(r1 < r2 + MIN_DIFFERENCE &&
        r1 > r2 - MIN_DIFFERENCE) {
        return dag->preferred_parent;
    } else if(r1 < r2) {
        return p1;
    } else {

```

```
    return p2;
}
}

static void
update_metric_container(rpl_dag_t *dag)
{
    dag->mc.type = RPL_DAG_MC_NONE;
}
```

Anexo B - Código para execução da Função Objetivo MRHOF

```
static void
reset(rpl_dag_t *sag)
{
    PRINTF("RPL: Reset MRHOF\n");
}

static void
neighbor_link_callback(rpl_parent_t *p, int status, int numtx)
{
    uint16_t recorded_etx = p->link_metric;
    uint16_t packet_etx = numtx * RPL_DAG_MC_ETX_DIVISOR;
    uint16_t new_etx;

    /* Do not penalize the ETX when collisions or transmission errors occur. */
    if(status == MAC_TX_OK || status == MAC_TX_NOACK) {
        if(status == MAC_TX_NOACK) {
            packet_etx = MAX_LINK_METRIC * RPL_DAG_MC_ETX_DIVISOR;
        }

        new_etx = ((uint32_t)recorded_etx * ETX_ALPHA +
                   (uint32_t)packet_etx * (ETX_SCALE - ETX_ALPHA)) / ETX_SCALE;

        PRINTF("RPL: ETX changed from %u to %u (packet ETX = %u)\n",
               (unsigned)(recorded_etx / RPL_DAG_MC_ETX_DIVISOR),
               (unsigned)(new_etx / RPL_DAG_MC_ETX_DIVISOR),
               (unsigned)(packet_etx / RPL_DAG_MC_ETX_DIVISOR));
        p->link_metric = new_etx;
    }
}

static rpl_rank_t
calculate_rank(rpl_parent_t *p, rpl_rank_t base_rank)
{
    rpl_rank_t new_rank;
    rpl_rank_t rank_increase;

    if(p == NULL) {
        if(base_rank == 0) {
            return INFINITE_RANK;
        }
        rank_increase = RPL_INIT_LINK_METRIC * RPL_DAG_MC_ETX_DIVISOR;
    } else {
        rank_increase = p->link_metric;
        if(base_rank == 0) {
            base_rank = p->rank;
        }
    }
}
```

```

if(INFINITE_RANK - base_rank < rank_increase) {
    /* Reached the maximum rank. */
    new_rank = INFINITE_RANK;
} else {
    /* Calculate the rank based on the new rank information from DIO or
       stored otherwise. */
    new_rank = base_rank + rank_increase;
}

return new_rank;
}

static rpl_dag_t *
best_dag(rpl_dag_t *d1, rpl_dag_t *d2)
{
    if(d1->grounded != d2->grounded) {
        return d1->grounded ? d1 : d2;
    }

    if(d1->preference != d2->preference) {
        return d1->preference > d2->preference ? d1 : d2;
    }

    return d1->rank < d2->rank ? d1 : d2;
}

static rpl_parent_t *
best_parent(rpl_parent_t *p1, rpl_parent_t *p2)
{
    rpl_dag_t *dag;
    rpl_path_metric_t min_diff;
    rpl_path_metric_t p1_metric;
    rpl_path_metric_t p2_metric;

    dag = p1->dag; /* Both parents are in the same DAG. */

    min_diff = RPL_DAG_MC_ETX_DIVISOR /
                PARENT_SWITCH_THRESHOLD_DIV;

    p1_metric = calculate_path_metric(p1);
    p2_metric = calculate_path_metric(p2);

    /* Maintain stability of the preferred parent in case of similar ranks. */
    if(p1 == dag->preferred_parent || p2 == dag->preferred_parent) {
        if(p1_metric < p2_metric + min_diff &&
           p1_metric > p2_metric - min_diff) {
            PRINTF("RPL: MRHOF hysteresis: %u <= %u <= %u\n",
                   p2_metric - min_diff,
                   p1_metric,
                   p2_metric + min_diff);
            return dag->preferred_parent;
        }
    }

    return p1_metric < p2_metric ? p1 : p2;
}

#if RPL_DAG_MC == RPL_DAG_MC_NONE
static void
update_metric_container(rpl_instance_t *instance)

```

```
{
    instance->mc.type = RPL_DAG_MC;
}
#else
static void
update_metric_container(rpl_instance_t *instance)
{
    rpl_path_metric_t path_metric;
    rpl_dag_t *dag;
#if RPL_DAG_MC == RPL_DAG_MC_ENERGY
    uint8_t type;
#endif

    instance->mc.type = RPL_DAG_MC;
    instance->mc.flags = RPL_DAG_MC_FLAG_P;
    instance->mc.aggr = RPL_DAG_MC_AGGR_ADDITIVE;
    instance->mc.prec = 0;

    dag = instance->current_dag;

    if (!dag->joined) {
        PRINTF("RPL: Cannot update the metric container when not joined\n");
        return;
    }

    if(dag->rank == ROOT_RANK(instance)) {
        path_metric = 0;
    } else {
        path_metric = calculate_path_metric(dag->preferred_parent);
    }

#if RPL_DAG_MC == RPL_DAG_MC_ETX
    instance->mc.length = sizeof(instance->mc.obj.etx);
    instance->mc.obj.etx = path_metric;

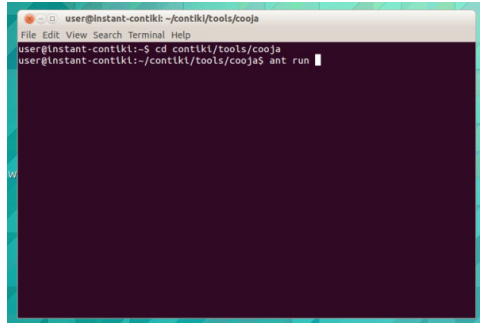
    PRINTF("RPL: My path ETX to the root is %u.%u\n",
           instance->mc.obj.etx / RPL_DAG_MC_ETX_DIVISOR,
           (instance->mc.obj.etx % RPL_DAG_MC_ETX_DIVISOR * 100) /
           RPL_DAG_MC_ETX_DIVISOR);
#elif RPL_DAG_MC == RPL_DAG_MC_ENERGY
    instance->mc.length = sizeof(instance->mc.obj.energy);

    if(dag->rank == ROOT_RANK(instance)) {
        type = RPL_DAG_MC_ENERGY_TYPE_MAINS;
    } else {
        type = RPL_DAG_MC_ENERGY_TYPE_BATTERY;
    }

    instance->mc.obj.energy.flags = type << RPL_DAG_MC_ENERGY_TYPE;
    instance->mc.obj.energy.energy_est = path_metric;
#endif /* RPL_DAG_MC == RPL_DAG_MC_ETX */
}
#endif /* RPL_DAG_MC == RPL_DAG_MC_NONE */
```

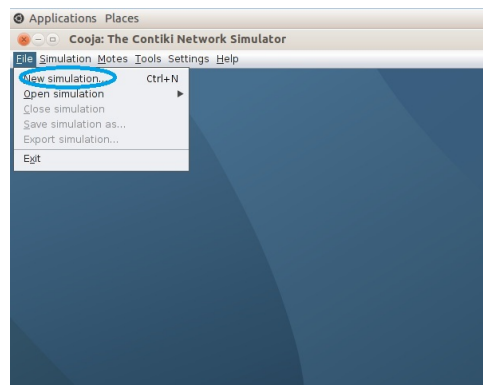
Anexo C - Algumas etapas para inicialização do processo de simulação RPL

Figura 35: Inicialização da rede



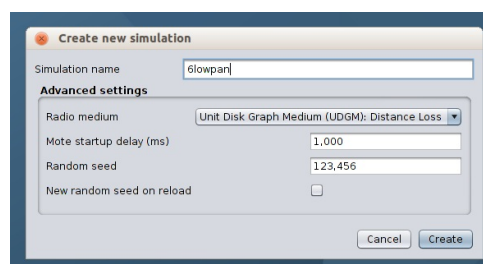
Fonte: Simulador Cooja

Figura 36: Iniciando uma simulação



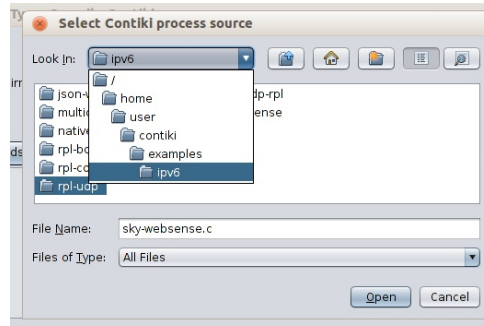
Fonte: Simulador COOJA

Figura 37: Parâmetros e criação da simulação



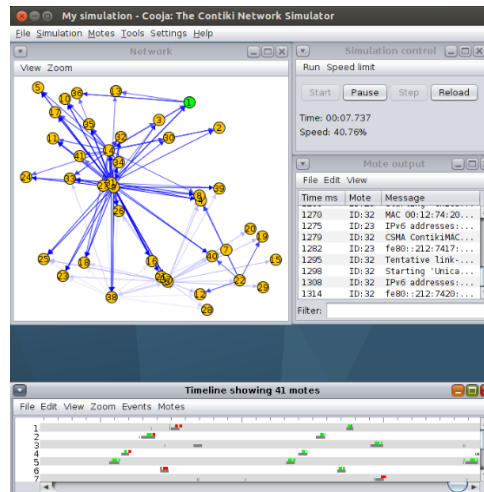
Fonte: Simulador COOJA

Figura 38: Processo de Simulação



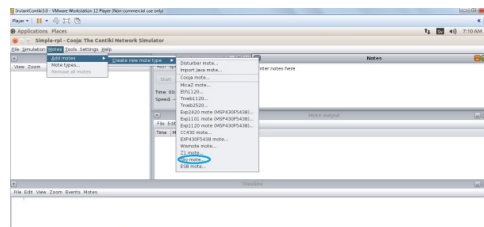
Fonte: Simulador COOJA

Figura 39: Simulação da rede e paine principal



Fonte: Simulador COOJA

Figura 40: Simulação da rede RPL



Fonte: Simulador COOJA