

FABIO COCCHI DA SILVA EIRAS

Medição da Precisão do Sincronismo em Redes de Sensores Móveis

Versão Original

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências

São Paulo
(2019)

FABIO COCCHI DA SILVA EIRAS

Medição da Precisão do Sincronismo em Redes de Sensores Móveis

Versão Original

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências

Área de Concentração: Sistemas
Eletrônicos

Orientador: Prof. Dr. Wagner Luiz Zucchi

São Paulo
(2019)

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Eiras, Fabio Cocchi da Silva
Medição da Precisão do Sincronismo em Redes de Sensores Móveis / F. C.
S. Eiras -- São Paulo, 2019.
161 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo.
Departamento de Engenharia de Sistemas Eletrônicos.

1.Sincronização 2.Telecomunicações 3.Wireless 4.Redes de Sensores
Móveis 5.Probabilidade de Cobertura e Comunicação I.Universidade de São
Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas
Eletrônicos II.t.

*Dedico este trabalho à minha amada esposa Beatriz e ao meu filho Rafael, na
esperança de ter contribuído de alguma forma para melhorar suas vidas.*

AGRADECIMENTOS

Agradeço ao meu orientador, Professo Dr. Wagner Luiz Zucchi, pela clareza, objetividade, generosidade e infindáveis horas de debates dedicadas ao encaminhamento de todas as opções e dúvidas surgidas ao longo da pesquisa.

À minha irmã Fernanda Eiras Geraque pela elaboração da maioria das figuras contidas neste trabalho.

Aos meus pais por sempre me apoiarem e me incentivarem a continuar com os estudos e com a busca contínua por conhecimento.

Finalmente, fica a minha gratidão especial à minha amada esposa Beatriz e ao meu filho Rafael, que, além de dedicarem valioso tempo, muito amor e paciência na colaboração permanente à realização deste trabalho, permitiram que eu muitas vezes dedicasse o tempo da família para a execução destes estudos.

“Quem nunca errou nunca experimentou algo novo.”

(Albert Einstein, 1879-1955)

RESUMO

O rápido desenvolvimento dos drones juntamente com a evolução das redes de sensores proporcionaram o aumento e a variabilidade de aplicações destas tecnologias, tanto no campo civil quanto militar, quando utilizadas de forma conjunta. Entre elas pode-se citar: sensoriamento agrícola, monitoramento de linhas de distribuição de energia, monitoramento de tráfego rodoviário e ferroviário e até mesmo aplicações de transporte de passageiros ou carga. A característica comum entre estas aplicações é o sensoriamento remoto executado por mais de um nó, normalmente móvel. Quando este sensoriamento é implementado com o uso de drones é necessário o estabelecimento de uma rede de comunicação entre eles. Esta rede é chamada de *Flying Ad Hoc Networks* (FANETs). Devido a características específicas deste tipo de rede tais como: velocidade de deslocamento dos nós sensores, mudança constante de topologia e distâncias dos enlaces de comunicação entre outras se faz necessária a análise de diversos aspectos técnicos entre eles o sincronismo. Dependendo da aplicação que utilizará os dados coletados pelos sensores móveis, é necessário que o sincronismo desta rede possua maior precisão de forma que a informação coletada pelos nós possa ser correlacionada de forma adequada, cumprindo assim os requisitos da aplicação. Desta forma este trabalho diferentemente de outros estudos apresenta um modelo de análise de sincronismo baseado em troca de mensagens que permite a avaliação da precisão de sincronismo da rede de sensores baseado nos requisitos da aplicação, considerando como variáveis a área de sensoriamento, a capacidade de transmissão e recepção dos nós sensores, a cobertura da área de sensoriamento, a probabilidade de conectividade dos nós sensores distribuídos e a velocidade de deslocamento dos nós sensores. Através de simulações desenvolvidas a partir de um modelo estatístico implementado em Simulink® é possível concluir a viabilidade de se utilizar sincronismo baseado em troca de mensagens em redes de sensores para se obter medidas da ordem de nanossegundos.

Palavras-Chave: Sincronização. Telecomunicações. Wireless. Redes de Sensores Móveis. Probabilidade de Cobertura e Comunicação.

ABSTRACT

The rapid development of drones, along with the evolution of sensor networks, has increased the use and variability of civilian and military applications of this technology, when used together. These include: agricultural sensing, power distribution line monitoring, climate monitoring and highway and railway monitoring, and even applications for passenger or cargo transportation. The common characteristic among these applications is remote sensing performed by more than one node, usually mobile. When this sensing is deployed with the use of drones, a communication network among them is required. This network is called *Flying Ad Hoc Networks* (FANETs). Due to the specific characteristics of this kind of network, such as the dislocation speed of the sensor nodes, the constant changes in topology and the distances from the communication links, to name a few, many different technical aspects need to be analyzed. One of these aspects is synchronization. Depending on the application that will use the data collected by the mobile sensors, the synchronization of the network must be very precise, or order to ensure that the information collected by the nodes is adequately correlated, thus fulfilling the requirements of the application. Therefore, this thesis, unlike previous studies, presents a model for analyzing synchronization based on the exchanging of messages. This allows the precision of the sensor network synchronism to be analyzed based on the requirements of the application, considering the variables of sensing area, the transmission and receiving capacity of the sensor nodes, the sensing area coverage, the connectivity probability of the distributed sensor nodes and the dislocation speed of the sensor nodes. By means of simulations developed using a statistical model deployed in Simulink®, it is possible to determine the feasibility of using synchronism based on message exchanging in sensor networks, in order to obtain measurements in the order of nano-seconds.

Keywords: Synchronism. Telecommunication. Wireless. Mobile Mobile Sensor Network. Probabilty of Coverage and Communication.

LISTA DE ILUSTRAÇÕES

Figura 1 – Possíveis utilizações de DRONES em FANETs	22
Figura 2 - Detecção Radar e Fusão de Dados.....	34
Figura 3 - Composição de um Rede de Sensores	40
Figura 4 - Hierarquia de Sincronização em Redes	48
Figura 5 - IEEE 1588 - Mecanismo de Sincronismo	50
Figura 6 - Exemplo de Relógio Transparente com Atraso	52
Figura 7 - Medição do Atraso Ponto-a-Ponto.....	53
Figura 8 - Erro de Sincronismo	55
Figura 9 - Modelo Comparador de Pulsos	60
Figura 10 - Diferença de Tempo entre Geradores de Pulso	60
Figura 11 - Diferença de Frequência.....	61
Figura 12 - Modelo com Ajuste Automático de Frequência	61
Figura 13 - Simulação com Ajuste Automático de Relógio	62
Figura 14 - Simulador Alto Nível	63
Figura 15 - Diagrama Alto nível.....	65
Figura 16 - Modelo de Transição de Estados	69
Figura 17 - Distribuição dos Nós em uma Área de Sensoriamento	71
Figura 18 - Deslocamento do nó Sensor 3 no Instante t_0 e t_1	73
Figura 19 - Representação do Alcance de Transmissão dos Nós Sensores.....	74
Figura 20 - Probabilidade de Recebimento de Mensagens	75
Figura 21 - Área de Alcance de Transmissão do Nó Sink	77
Figura 22 - Área de Alcance de Transmissão Total (Nó Sink + Nó X ₁).....	78
Figura 23 - Área adicionada pelo Nó X ₁ ao Alcance do Nó Sink.....	79
Figura 24 - Possível Posicionamento do nó X ₁ - Casos (a) e (b).....	81
Figura 25 - Triângulo Formado pela Intersecção de Três Círculos.....	83
Figura 26 - Possível Posicionamento do nó X ₁ – Caso (c).....	84
Figura 27 - Possível Posicionamento do nó X ₁ – Caso (d).....	85
Figura 28 - Ordem do Ângulo α	86
Figura 29 - Diagrama de Blocos em Alto Nível	90
Figura 30 - Componente Message Loss	91
Figura 31 - Componente Network Delay.....	92

Figura 32 - Geração de Pulsos Aleatórios na Escala de Tempo do Modelo.....	93
Figura 33 - Bloco Correction Module.....	94
Figura 34 - Validação de Comunicação entre dois nós sensores.....	96
Figura 35 - Lógica do Teste de Comunicação.....	97
Figura 36 - Probabilidade de ocorrência do Evento $E_3 (X_2 \rightarrow X_1 \rightarrow S)$	100
Figura 37 - Gerador de Pulso de Controle.....	101
Figura 38 - Cálculo MTIE - Perda 100%.....	104
Figura 39 - Atraso entre Master e Slave - Perda 100%.....	104
Figura 40 - Cálculo MTIE - Sem Perda.....	107
Figura 41 - Atraso entre Master e Slave – Sem Perda.....	107
Figura 42 - Taxa Média de Perda x Quantidade de Nós Sensores.....	110
Figura 43 - Quantidade de Sensores Ativos x Taxa média de Perda – $r=0,1R$	112
Figura 44 – Taxa Média de Perda em Função de r/R	113
Figura 45 - Gráfico MTIE.....	115
Figura 46 - Parâmetro MTIE - Intervalo de Taxa de Perda [7,04% - 2,41%].....	117
Figura 47 - Parâmetro TDEV.....	119
Figura 48 – Histogramas.....	120
Figura 49 - MTIE Diferença de Frequência.....	122
Figura 50 – Simulador.....	148
Figura 51 - Gráfico MTIE - 5000 / 6000 / 7000 / 7300 sensores.....	158
Figura 52 - Gráfico MTIE - 7700 / 8500 / 9000 sensores.....	159

LISTA DE TABELAS

Tabela 1 - Transição dos Estados.....	88
Tabela 2 - Eventos Possíveis.....	98
Tabela 3 - Parâmetros de Validação.....	98
Tabela 4 - Probabilidade por Evento – Calculado x Simulado	99
Tabela 5 - Parâmetros de Simulação – Vetor de Perda.....	110
Tabela 6 - Parâmetros de Simulação - Relação r/R.....	112
Tabela 7 - Parâmetros de Simulação - Precisão de Sincronismo.....	114
Tabela 8 - Parâmetros de Simulação - Precisão de Sincronismo – Cenário 2	117
Tabela 9 - MTIE e Perda - Cálculo TDEV	118
Tabela 10 - Diferença de Frequência Master / Slave.....	122
Tabela 11 - Vetor de Perda - Intervalo de 4000 a 7300 nós	149
Tabela 12 - Vetor de Perda - Intervalo de 7500 a 10000 nós	153

LISTA DE SIGLAS

ADEV	<i>Allan Deviation</i>
DECEA	<i>Departamento de Controle do Espaço Aéreo</i>
EUROCONTROL	<i>European Organization for the Safety of Air Navigation</i>
eVTOL	<i>Electrical Vertical Take-off and Landing</i>
FAA	<i>Federal Aviation Administration</i>
FANET	<i>Flying Ad Hoc Networks</i>
FDP	<i>Função densidade de distribuição de probabilidade</i>
FLIR	<i>Forward Looking Infra Red</i>
FTSP	<i>Flooding Time Synchronizaton Protocol</i>
GCS	<i>Ground Control Station</i>
GPS	<i>Global Positioning System</i>
GTSP	<i>Gradient Time Synchronization Protocol</i>
HS	<i>Harmony Search Algorithm</i>
ICAO	<i>International Civil Aviation Organization</i>
IEEE	<i>Institute of Electrical and Electronics Engineering</i>
ITU-T	<i>International Telecommunication Union – Telecommunication Standardization Sector</i>
MAC	<i>Media Access Control</i>
MANET	<i>Mobile Ad hoc Networks</i>
MDEV	<i>Modified Allan Deviation</i>
MTIE	<i>Maximum Time Interval Error</i>
NOAA	<i>National Oceanic and Atmospheric Administration</i>
ns	<i>Nano-segundos</i>
NTP	<i>Network Time Protocol</i>
PRC	<i>Primary Reference Clock</i>
PSO	<i>Particle Swarm Optimization</i>
PTP	<i>Precision Time Protocol</i>
QoS	<i>Quality of Service</i>
RBS	<i>Reference Broadcast Synchronization</i>
SDH	<i>Synchronous Digital Hierarchy</i>

SyncE	<i>Synchronous Ethernet</i>
TDEV	<i>Time Deviation</i>
TDM	<i>Time Division Multiplexing</i>
TIErms	<i>Root Mean Square of Time Interval Error</i>
TPSN	<i>Timing-Sync Protocol for Sensor Networks</i>
V2V	<i>Vehicle to Vehicle</i>
V2R	<i>Vehicle to Road</i>
VANETs	<i>Vehicular Ad Hoc Networks</i>
VANTs	<i>Veículos Aéreos não Tripulados</i>
WMN	<i>Wireless Mesh Network</i>

LISTA DE SÍMBOLOS

A_{SR}	Área de Sensoriamento
D_p	Distância Percorrida pelo nó sensor em um intervalo de tempo de simulação
K_1	Relação entre N_1 e N_2 (N_1 / N_2)
K_2	probabilidade de um determinado nó sensor receber a mensagem de sincronismo dado a sua mobilidade
N_1	é a quantidade total de nós sensores distribuídos de forma aleatória dentro do perímetro da área de sensoriamento.
N_2	N_2 é a quantidade de nós sensores HABILITADOS dentro do perímetro da área de sensoriamento, portanto é um subconjunto de N_1 com a relação $N_2 \leq N_1$
R	Raio da Área de Sensoriamento
r	Raio de alcance de transmissão do nós sensor
V_d	Velocidade de deslocamento de um nó sensor
β_1	probabilidades de transição entre os estados. De <i>HABILITADO</i> para <i>NÃO HABILITADO</i>
β_2	probabilidades de transição entre os estados. De <i>NÃO HABILITADO</i> para <i>HABILITADO</i>
λ_1	probabilidades de transição entre os estados de <i>HABILITADO</i> para <i>FORA DE ALCANCE</i>
λ_2	probabilidades de transição entre os estados. De <i>FORA DE ALCANCE</i> para <i>HABILITADO</i>

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Trabalhos Relacionados.....	23
1.2	Objetivo	31
1.3	Motivação	32
1.4	Metodologia.....	36
2	REDES DE SENSORES.....	39
2.1	FANET – Flying Ad hoc Networks.....	42
2.1.1	Pontos em Aberto.....	43
3	SINCRONIZAÇÃO PRECISA DE TEMPO.....	45
3.1	Sincronização em uma Rede Wireless Mesh.....	46
3.2	Sincronização em Redes de Pacotes	46
3.2.1	Protocolo IEEE 1588	49
3.2.1.1	Extensão do Modelo para Atraso Assimétrico	51
3.3	Medição de Sincronismo	54
3.3.1	<i>Allan Deviation</i> – ADEV e <i>Modified Allan Deviation</i> (MDEV)	55
3.3.2	<i>Time Deviation</i> (TDEV)	57
3.3.3	<i>Maximum Time Interval Error</i> (MTIE)	59
3.3.4	<i>Simulink</i> ® como Ferramenta para Aferição do Tempo	59
4	DESENVOLVIMENTO DO MODELO	64
4.1	Visão Geral.....	64
4.2	Composição do Modelo.....	66
4.2.1	Nós Sensores.....	66
4.2.2	Sincronismo.....	68
4.3	Transição de Estados.....	69
4.4	Modelagem do Período de Repouso.....	70

4.5	Modelagem da Mobilidade	72
4.5.1	Caso 1 – Alta Velocidade de Deslocamento do Nó Sensor	73
4.5.2	Caso 2 – Baixa Velocidade de Deslocamento do Nó Sensor	75
4.5.3	Cálculo de P_{random}	77
4.5.3.1	Generalização de P_{random}	81
4.5.4	Metodologia para Cálculo do Parâmetro K2	86
4.6	Algoritmo de Transição de Estados	87
5	IMPLEMENTAÇÃO DO SIMULADOR	90
5.1	Descrição do Simulador	90
5.1.1	Componente Master	91
5.1.2	Componente Message Loss	91
5.1.3	Componente Network Delay	92
5.1.4	Componente Slave	93
5.2	Modo de Operação do Modelo	94
5.2.1	Script de Simulação	94
5.2.1.1	Validação do Script de Simulação	98
5.2.2	Funcionamento do Simulador	100
5.3	Validação do Simulador	103
5.3.1	Validação com Perda	103
5.3.2	Validação sem Perda	105
6	SIMULAÇÕES E RESULTADOS	109
6.1	Geração do Vetor de Perda	109
6.2	Análise da Relação Raio de Alcance x Raio da Área de Sensoriamento	111
6.3	Simulação Precisão de Sincronismo	113
6.3.1	Cenário 1 – Precisão do Sincronismo em Função da Quantidade de Nós Sensores	114

6.3.2	Cenário 2 – Precisão do Sincronismo em Função da Quantidade de Nós Sensores – Taxa de Perda de 5%	116
6.3.2.1	Análise da Estabilidade do Sincronismo	118
6.3.3	Cenário 3 – Variação da Frequência Master / Slave.....	121
7	CONSIDERAÇÕES FINAIS	124
7.1	Contribuições	124
7.2	Trabalhos futuros	126
	REFERÊNCIAS.....	127
	APÊNDICE A – RESOLUÇÃO EQUAÇÃO	133
	APÊNDICE B – SCRIPTS MATLAB.....	141
	APÊNDICE C – IMPLEMENTAÇÃO SIMULINK®.....	147
	APÊNDICE D – VETORES DE PERDA.....	149
	APÊNDICE E – GRÁFICOS MTIE	158

1 INTRODUÇÃO

Este trabalho tem como tema principal a medição da precisão de sincronismo entre relógios, a forma de distribuição desta informação utilizando troca de mensagens baseada em comutação de pacotes e como isto impacta as redes de sensores levando em consideração as características das aplicações para as quais estas redes foram implementadas.

As redes de sensores são caracterizadas pela distribuição de nós sensores capazes de realizar a captura de algum tipo de informação, por exemplo temperatura, posição, profundidade, velocidade de vento ou até mesmo captura de imagem e pela transmissão destas informações para um sistema capaz de interpretar e processar os dados coletados. Os sensores podem ser distribuídos em posições previamente estabelecidas ou de forma aleatória em uma área determinada, chamada de área de sensoriamento, e também podem ser fixos ou móveis, de acordo com a aplicação.

Nos dias atuais as redes de sensores possuem as mais diversas aplicações. No campo das aplicações civis pode-se destacar: sensoriamento agrícola, monitoramento de rebanhos, monitoramento de linhas de distribuição de energia, monitoramento de clima e monitoramento de tráfego rodoviário e ferroviário. No campo militar pode-se destacar: sensoriamento remoto de fronteiras, monitoramento do espaço aéreo, da costa marítima, do território inimigo e posicionamento de tropa.

Ainda dentro das aplicações de sensoriamento remoto, a utilização de nós sensores móveis têm crescido. Isto ocorre principalmente devido à evolução destes equipamentos no que tange à capacidade de processamento, memória, capacidade de recepção e envio de dados e baixo consumo de bateria. Este tipo de sensoriamento é frequentemente aplicado em cenários de desastres causados pela natureza ou não, permitindo que a força tarefa envolvida no resgate de vítimas tenha informações mais completas e em tempo real, sem aumentar o risco de exposição da equipe de resgate, bem como em missões de segurança civil ou militar com o propósito de

reconhecimento, vigilância e defesa. Desta forma as decisões de como devem ser estabelecidos os procedimentos de resgate se tornam mais assertivas e rápidas.

No Brasil os potenciais usuários de aplicações de sensoriamento são os Governos Municipais, Estaduais e o Governo Federal, através dos mais variados órgãos e departamentos, tais como Defesa Civil, Força Aérea Brasileira, Ministério da Defesa, Comando da Marinha e Comando do Exército entre outros e empresas concessionárias de serviços de energia, água e telecomunicações. Apesar de aplicações logísticas e de transporte de passageiros através de drones, serem exploradas por empresas do setor privado, a utilização do espaço aéreo é regulada por órgãos governamentais, e, portanto, estas empresas dependem de setores do governo para operarem este tipo de serviço.

Entretanto, apesar de ser uma tecnologia acessível para estes usuários, sua implementação e consequente utilização ainda é baixa e restrita a atividades muito específicas como por exemplo monitoramento de tráfego aéreo. Para efeito de comparação e entendimento de como o Brasil subutiliza este tipo de tecnologia pode-se comparar a quantidade de sensores meteorológicos instalados no Brasil e nos Estados Unidos, dois países que possuem dimensões continentais. Enquanto no Brasil, segundo o site Portal Brasil (PORTAL BRASIL, 2014) o governo federal licitou a compra de 9 sensores para serem integrado à base existente de 20 radares, totalizando 29 sensores meteorológicos instalados em solo nacional o *National Oceanic and Atmospheric Administration* (NOAA, 2016), órgão ligado ao Departamento de Comércio dos Estados Unidos, possui uma rede de sensores composta por 160 sensores meteorológicos, o que representa uma diferença de aproximadamente cinco vezes e meia. É importante ressaltar que estes números não consideram sensores pertencentes a universidades e empresas privadas, mas já demonstram o potencial de crescimento do uso de tecnologias de sensoriamento no Brasil.

Dois exemplos demonstram isto: o primeiro foram os deslizamentos ocorridos no ano de 2011 na região serrana do estado do Rio de Janeiro. Se uma rede de sensores meteorológicos compostos basicamente por radares meteorológicos para previsão de curto e curtíssimo prazo, sensores de monitoramento de taludes e encostas e estações meteorológicas de superfície estivesse disponível na região, os deslizamentos ocorridos teriam sido menos catastróficos, isto é, vidas teriam sido

salvas e o prejuízo financeiro tanto da população quanto do estado teriam sido em menor escala pois seria possível alertar a população e coordenar as atividades de defesa civil, bombeiros e hospitais com antecedência. O segundo exemplo, foi o rompimento da barragem de detritos minerais em Mariana, Minas Gerais, ocorrido em 2014 e mais recentemente o rompimento da barragem de Brumadinho, também em Minas Gerais, ocorrido em 2019. Para ambos os casos tanto o sensoriamento da barragem de forma preventiva, quanto o sensoriamento remoto efetuado por sensores móveis, como por exemplo drones, após o rompimento da barragem não foram utilizados. Isto resultou em um tempo de resposta elevadíssimo, acarretando danos imensos à população, ao meio-ambiente e às cidades da região (AZEVEDO, 2019). Um estudo realizado na Índia (RAMESH, 2014) demonstra a viabilidade do monitoramento do solo com o objetivo de prever antecipadamente a ocorrência de um deslizamento de terra e conseqüentemente diminuir os impactos causados por este tipo de catástrofe na sociedade.

Uma forma de implementar o sensoriamento remoto é através da utilização de drones, também chamados de Veículos Aéreos não Tripulados (VANTs). Considerando a utilização de sensores em drones pode-se citar outras aplicações de grande porte e criticidade, que podem se beneficiar desta capacidade de sensoriamento, mobilidade e conectividade, as quais estão ganhando relevância, e que também demandam coordenação precisa, bem como possuem critérios rígidos de posicionamento dos nós, seja em relação ao ambiente físico, seja em relação a outros nós pertencentes à rede. São elas: aplicações logísticas para a realização de entregas em grandes centros urbanos através do uso de drones, a utilização de drones autônomos de grande porte, conhecidos como *Electrical Vertical Take-off and Landing* (eVTOL) para transporte de passageiros e até mesmo a coordenação de espaçonaves para exploração do universo e outros planetas (ELY et al., 2018).

Vale ressaltar que sob este aspecto é possível que um drone tenha seu voo coordenado de forma autônoma ou tenha o seu voo coordenado por um piloto remoto. Para a realização de um voo autônomo, independente da aplicação, é necessário que o drone possua diversos sensores para que ele possa “sentir o ambiente” em que está voando e assim corrigir sua trajetória sem a necessidade de uma intervenção humana. Já para o voo remotamente pilotado o drone também estará equipado com diversos sensores, porém a decisão final do movimento a ser executado pelo drone será do

piloto remoto. Neste caso pode-se considerar que o drone encontra-se totalmente pilotado, isto é, controlado por um ser humano ou está em um voo semiautônomo, isto é, somente em situações críticas, as quais são alertadas pelo algoritmo de controle do drone é que o piloto intervirá.

Existem dois tipos de drones: os de asa fixa e os de asa rotativa. As principais diferenças entre eles são:

- 1- As aeronaves de asa rotativa podem permanecer estacionárias em uma determinada posição;
- 2- As aeronaves de asa rotativa possuem menor capacidade de carga do que as de asa fixa. Isto permite que as aeronaves de asa fixa possam carregar uma quantidade maior de sensores;
- 3- As aeronaves de asa fixa possuem maior autonomia do que as de asa rotativa.

Uma vez que exista mais de um drone para executar o sensoriamento é possível formar uma rede de comunicação entre eles. Esta rede de comunicação tem como objetivo a transmissão de informação capturada e/ou comando e controle dos drones e é essencialmente uma Rede Móvel *Ad hoc* – *Mobile Ad hoc Networks* (MANET). No entanto, devido às características específicas dos nós móveis, neste caso os drones, como velocidade, mobilidade, qualidade do serviço, a descoberta de outros nós e a capacidade de entrega dos dados capturados, esta rede tornou-se conhecida como *Flying Ad Hoc Network* (FANET) (SAHINGOZ, 2013).

A Figura 1 apresenta três possíveis cenários para utilização de drones e FANET, explicados a seguir:

Cenário 1 – os próprios drones são equipados com um ou mais tipos de sensores. Para este cenário o drone pode ser autônomo ou não, e diversas aplicações podem ocorrer neste cenário. Entre elas: sensoriamento de terrenos; transporte de carga e/ou passageiros e sensoriamento de território inimigo para coordenação de ataques. Dependendo da aplicação é possível que cada drone possua um tipo diferente de sensor e desta forma, a informação completa será composta pela agregação dos diversos sensores existentes. Esta composição ocorre na estação base de solo, chamada de *Ground Control Station* (GCS).

Cenário 2 – Utilizar os drones para formar uma infraestrutura de comunicação em ambientes de catástrofes, onde a infraestrutura local está comprometida. Desta forma é possível prover comunicação entre as diversas entidades envolvidas e os membros das equipes de resgate.

Cenário 3 – Os drones são utilizados para capturar a informação coletada por sensores espalhados em uma determinada área. Aplicações como sensoriamento florestal ou agrícola se adequam a este tipo de cenário. Neste caso os drones não funcionam como sensores e são apenas agentes de transmissão de dados. Usualmente para este cenário, os drones são remotamente pilotados ou possuem um plano de voo pré-estabelecido e carregado em seus sistemas e em caso de situação crítica o drone obrigatoriamente retorna a um ponto pré-estabelecido pelo responsável pela missão.

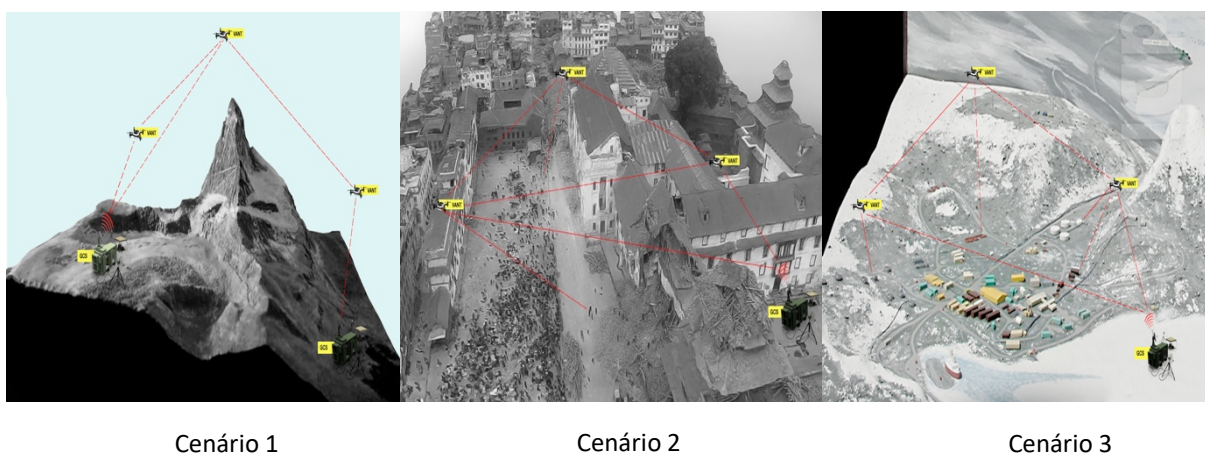


Figura 1 – Possíveis utilizações de DRONES em FANETs

Fonte: Elaborada pelo Autor

Sob o aspecto de comunicação, para os três cenários é necessário o estabelecimento de enlaces de comunicação tanto entre os diversos drones da missão como entre os drones e a GCS, formando assim uma FANET. Os principais objetivos desta rede de comunicação são:

- 1- Quando sob voo controlado, permitir que a GCS se comunique com os drones e assim efetue o controle de voo destes equipamentos;
- 2- Permitir que as informações de sensoriamento coletadas pelos drones sejam transmitidas para a GCS sempre que uma conexão estiver disponível;

- 3- Permitir encaminhamento de informações entre os nós (drones), não necessitando exclusivamente de enlaces com visada direta entre todos os drones e a GCS. Estas informações podem ser de dados de monitoramento coletados ou dados do ambiente do voo para coordenação de voo autônomo.
- 4- Prover infraestrutura de comunicação entre membros de uma equipe.

Apesar da utilização de drones e FANET, conforme citado acima, ser vantajosa sob o ponto de vista de custo, mobilidade, facilidade de transporte, rapidez de disponibilização de infraestrutura e diminuição da exposição de pessoas em áreas de risco, existem diversos desafios a serem vencidos. Entre eles pode-se citar a confiabilidade dos enlaces de comunicação, a capacidade de roteamento, o consumo de energia, perda de pacotes, atraso, interferências e até mesmo a não permissão de coexistência de drones e aeronaves tripuladas no mesmo espaço aéreo.

As FANETs podem sofrer diversos tipos de interferência, provocados por diversas fontes tais como clima, relevo, prédios e lagos. Isto faz com que este tipo de rede tenha que possuir mecanismos adaptativos que permitam que a rede continue disponível durante todo o tempo da missão. Um dos pontos críticos para isto é o estabelecimento de sincronismo entre todos os elementos envolvidos tanto drones como GCS. Em qualquer um dos três cenários propostos o sincronismo é necessário: para o cenário 1, se não houver sincronismo as informações coletadas pelos diversos drones não poderá ser recomposta ou será recomposta de forma imprecisa e no caso de transporte de carga e/ou passageiros a localização da aeronave e coordenação do voo autônomo poderá ser comprometida, para o cenário 2 a falta de sincronismo na rede pode acarretar perda de informação e para o último cenário, considerando que a FANET funcione como infraestrutura de comunicação, a transmissão de voz poderá ser impactada.

1.1 Trabalhos Relacionados

As redes de sensores têm sido exaustivamente estudadas sob diversos aspectos tais como consumo de energia, conectividade, otimização de distribuição de nós sensores com o objetivo de se obter a máxima cobertura com o mínimo de nós sensores, mobilidade e sincronismo.

Cobertura, conectividade e consumo de energia em uma rede de sensores estão fortemente correlacionados. A permanência no estado de dormência de muitos nós sensores aumenta a eficiência energética da rede como um todo porém reduz tanto a cobertura da área de sensoriamento, reduzindo a capacidade de detecção de eventos, como a conectividade da mesma (HIRANI; SINGH, 2015).

Uma determinada área está coberta se para todo e qualquer evento ocorrido dentro desta área existir um ou mais nós sensores capazes de detectá-lo. Em relação a conectividade pode-se considerar que uma rede de sensores sem fio está conectada se cada par de nós sensores for capaz de se comunicar direta ou indiretamente com outros nós sensores, com o objetivo de descobrir um subconjunto mínimo de nós sensores ativos para que os dados observados, dentro da área de sensoriamento, possam chegar ao sistema de processamento ou ao nó sensor sink (GUPTA; KUMAR; JAIN, 2016).

Segundo (WANG et al., 2017) a cobertura de uma rede de sensores pode tipicamente ser classificada de três formas: a) cobertura do evento - onde o objetivo é que quando ocorra um evento, o mesmo seja captado por um sensor; b) cobertura da área – tem como objetivo possuir o maior percentual possível de área sendo monitorada (coberta) por um sensor; c) cobertura por barreira – os sensores são posicionados de maneira a formar uma barreira, portanto todo e qualquer caminho de intrusão para a área de sensoriamento estará monitorado. Ainda segundo (WANG et al., 2017), o interesse de pesquisa sobre cobertura e conectividade em redes de sensores tem crescido recentemente, porém estes estudos tem focado em um único aspecto, por exemplo mobilidade (WANG; LIM; MA, 2009) e (SENOUCI et al., 2015), qualidade da cobertura ou modelo de instalação dos sensores (DEIF; GADALLAH, 2014).

O algoritmo *Particle Swarm Optimization* (PSO) é um algoritmo bastante utilizado em redes de sensores devido ao equilíbrio entre complexidade e qualidade da otimização (KULKARNI; MEMBER; KUMAR, 2011).

O trabalho desenvolvido em (AZIZ; MOHEMMED; ALIAS, 2009) utiliza o PSO para encontrar a distribuição dos sensores que oferece a melhor cobertura, enquanto o diagrama de Voronoi (AURENHAMMER, 1991) é usado para avaliar a adequação da solução. O algoritmo é avaliado através de simulação em diferentes cenários e os resultados da simulação sugerem que o algoritmo proposto consegue obter uma boa cobertura com melhor eficiência de tempo. Entretanto, devido à característica de

iteratividade do algoritmo o seu uso em aplicações de tempo real, que demandam otimizações frequentes, não é recomendado.

Em (ALIA; AL-AJOURI, 2017) o *Harmony Search Algorithm* (HS) é utilizado para melhorar a distribuição de sensores dentro de uma área de sensoriamento específica com o objetivo de identificar a quantidade ótima de sensores, assim como a melhor localização dos mesmos para maximizar a cobertura da rede de sensores. Os resultados obtidos pelos autores demonstram que a utilização do algoritmo HS proporciona uma melhora de 25% na cobertura quando comparado com a distribuição aleatória dos nós sensores. Entretanto simulações realizadas não consideram a conectividade da rede, o consumo de energia e a existência de obstáculos que podem interferir na distribuição dos nós sensores.

O estudo desenvolvido por Romoozi et al. (ROMOOZI et al., 2010) propõem uma otimização no consumo de energia enquanto preserva a cobertura da área de sensoriamento. Para tanto foi utilizado o *Genetic Algorithm* (HOLLAND, 1992) para melhor distribuição dos nós sensores e o *Fuzzy C-means Algorithm* (HATHAWAY; BEZDEK, 1994) para agrupar os nós em clusters. Os resultados obtidos demonstram um consumo de energia equilibrado enquanto se mantém a cobertura desejada.

O estudo desenvolvido por He et al. (HE et al., 2012) apresenta um modelo de sensores móveis, ao invés de sensores estáticos, usados para monitorar uma faixa da área de sensoriamento e formar assim uma espécie de barreira dinâmica. Para que este modelo tenha resultados satisfatórios, isto é, consiga de fato detectar eventos ocorridos ao longo da barreira é necessário que os sensores tenham conhecimento prévio da probabilidade de ocorrência de intrusão, assim é possível determinar a estratégia de mobilidade de cada sensor de forma mais assertiva. Porém para diversas aplicações, como por exemplo aplicações militares ou aplicações logísticas, não é possível obter este conhecimento prévio resultando assim em um baixo índice de detecção.

Em relação ao sincronismo, esta é uma variável que vem sendo estudada exaustivamente muito antes do advento das redes de sensores sem fio (SOMMER; WATTENHOFER, 2009). O crescimento do uso de redes de transmissão assíncrona de pacotes em aplicações de telecomunicações e de sensoriamento despertou o interesse sobre a possibilidade de transmissão de informações de sincronização através deste tipo de rede.

Cada vez mais, as redes padrão Ethernet vêm sendo utilizadas como *backhaul* das redes de telecomunicações substituindo os enlaces *Time Division Multiplex / Synchronous Digital Hierarchy* (TDM/SDH). Esta tendência justifica-se principalmente pelo menor custo dos enlaces padrão Ethernet que, em alguns casos, equivalem a 1/6 do valor dos enlaces SDH. Todavia, enlaces SDH podem fornecer um sinal de relógio de alta precisão adequado para operação das estações rádio base em telefonia celular e outras aplicações (ANYAEGBU; WANG; BERRIE, 2013).

A sincronização de tempo de elementos distribuídos é um requisito comum para diversos tipos de aplicações (CHO et al., 2009) entre as quais este trabalho foca nas aplicações em redes de sensores e em telecomunicações. A sincronização em redes de sensores sem fio é essencial para facilitar o desenvolvimento de aplicações colaborativas que realizam diversas operações tais como: localização, agregação de dados, amostragem distribuída, etc. (DEL RÍO et al., 2012).

Soluções tradicionais para se obter sincronismo como o uso de um relógio atômico, como o *Global Positioning System* (GPS) ou o uso do *Network Time Protocol* (NTP) possuem limitações (MILLS, 1991). No caso do GPS, este tipo de solução é viável para a maioria das aplicações das redes de sensores, porém algumas limitações são inerentes, principalmente no que se refere, a precisão, custos e consumo de energia (SOMMER; WATTENHOFER, 2009). Já para o NTP a principal limitação está na precisão, que é impactada principalmente pela assimetria dos enlaces de transmissão e pelos dispositivos receptores, tais como placas de rede ou até mesmo os servidores (NOVICK; LOMBARDI, 2015). Para tanto, diversos protocolos de sincronização voltados para este tipo de aplicação têm sido estudados, sendo alguns de uso para redes de telecomunicações e outros específicos para redes de sensores. Entre os protocolos de uso em telecomunicações pode-se citar o IEEE 1588 (IEEE, 2008), já entre os protocolos específicos para redes de sensores pode-se citar: *Reference Broadcast Synchronization* (RBS) (ELSON; GIROD; ESTRIN, 2002), *Timing-Sync Protocol for Sensor Networks* (TPSN) (GANERIWAL; KUMAR; SRIVASTAVA, 2003) e *Flooding Time Synchronization Protocol* (FTSP) (MARÓTI et al., 2004) e mais recentemente o *Gradient Time Synchronization Protocol* (GTSP) (SOMMER; WATTENHOFER, 2009).

O protocolo IEEE 1588 é um protocolo desenvolvido para redes de pacotes e é baseado em troca de mensagens. O trabalho desenvolvido por (WOBSCHALL; MA,

2010) utiliza o protocolo IEEE 1588 para analisar o sincronismo em uma rede de sensores baseada no protocolo IEEE 802.15.4 e não são consideradas variáveis como perda de conectividade dos sensores e velocidade de deslocamento. Os autores obtiveram um sincronismo de 10 μs quando o sistema é analisado em curtos períodos de tempo e de 100 μs quando analisado em longos períodos de tempo.

O protocolo RBS é baseado em broadcast para sincronizar um conjunto de sensores. Um nó de referência é eleito dentro de cada cluster e envia a mensagem para sincronizar todos os outros nós. O nó transmissor não envia uma mensagem com marca de tempo explícita. Cada nó receptor armazena o momento em que a mensagem de referência foi recebida de acordo com seu relógio local. Os nós receptores, então trocam estas observações entre si e a partir desta troca de mensagem os mesmos analisam as diferenças entre os tempos observados na recepção de cada mensagem e baseado em uma média destas observações atualizam seus relógios locais. Esta implementação permite que o RBS atinja uma precisão da ordem de microssegundos.

Durante a fase de descoberta do nível inicial, o algoritmo implementado no TPSN elege um nó sensor como principal e a partir deste nó constrói uma árvore de abrangência da rede de sensores. Na fase de sincronização do algoritmo, os nós são sincronizados com seus nós superiores hierárquicos na árvore através de troca de mensagens bidirecional. Usando as marcas de tempo incorporadas nas mensagens de sincronização, o nó receptor é capaz de calcular o atraso de transmissão e o deslocamento relativo do relógio. No entanto, o TPSN não compensa a variação de frequência do oscilador do relógio, o que torna obrigatória a re-sincronização frequente. Além disso, o TPSN causa uma alta sobrecarga de comunicação, pois uma troca de mensagens bidirecional é necessária para cada nó filho (SOMMER; WATTENHOFER, 2009). Quando comparado com o protocolo RBS o TPSN atinge um desempenho aproximadamente duas vezes melhor. As simulações realizadas por (GANERIWAL; KUMAR; SRIVASTAVA, 2003) obtiveram média de erro de 16.9 μs sendo que para o pior caso o erro foi de 44 μs para o TPSN, enquanto para o RBS os resultados foram 29.13 μs e 93 μs respectivamente.

O protocolo FTSP implementa um mecanismo que tem como objetivo minimizar as limitações dos protocolos anteriormente citados. A sincronização ocorre a partir de um

nó transmissor que envia uma mensagem com marca de tempo para vários nós receptores. Esta marca de tempo ocorre na camada *Media Access Control* (MAC) pois desta forma pode eliminar muitos dos erros. No entanto, a sincronização de tempo precisa em pontos discretos no tempo é apenas uma solução parcial. A compensação para o desvio de relógio dos nós é inevitável para alcançar pontos de sincronização entre pontos de alta precisão e para manter a sobrecarga de comunicação baixa. Para compensar o desvio do relógio o FTSP implementa a regressão linear. Como as redes de sensores normalmente operam em áreas maiores do que o alcance de um único salto de comunicação, o FTSP fornece sincronização de vários saltos. Para esta situação um nó raiz deve ser eleito, e periodicamente ele transmite uma mensagem que inunda a rede com seu registro de data e hora atual, formando assim uma estrutura de árvore ad hoc. A eleição do nó raiz é realizada de forma dinâmica com base no menor identificador de nó, isto significa que cada nó deve ter um identificador único. Após a inicialização, um nó aguarda algumas rodadas e escuta os *beacons* de sincronização de outros nós. Cada nó suficientemente sincronizado com o nó raiz inicia a transmissão de sua estimativa do relógio global. Se um nó não receber mensagens de sincronização durante um certo período, ele se autodeclarará como um novo nó raiz.

O protocolo GTSP implementa um algoritmo para sincronização onde os nós têm que concordar com um valor de relógio comum, que pode ser obtido calculando a média dos valores de relógio e é utilizado para a compensação de desvio. No entanto, usar a média de todos os nós vizinhos como o novo valor do relógio pode gerar distorções de sincronismo em situações em que os deslocamentos forem grandes. Um dos pontos restritivos deste algoritmo é que devido a estrutura de dados utilizada, o número máximo de vizinhos que um nó considera nos cálculos também é limitado e um nó poderá descartar informações essenciais do vizinho. Esta característica possui impacto mais significativo em redes mais densas o que pode inviabilizar sua utilização em aplicações de transporte de cargas e passageiros, uma vez que estes ambientes possuem densidades elevadas. Os resultados obtidos pelos autores demonstram um desempenho superior ao FTSP. Para os cenários simulados o erro médio de sincronismo obtido entre dois nós quando utilizado o protocolo GTSP, foi de $2.8 \mu\text{s}$ enquanto que para o mesmo cenário utilizando o protocolo FTSP o erro médio obtido foi de $15 \mu\text{s}$.

Se por um lado os protocolos apresentados estabelecem mecanismos de sincronismo em redes de sensores, por outro a evolução da tecnologia nos últimos anos em áreas como tecnologia embarcada, tamanho e capacidade de processadores e memórias e nas telecomunicações entre outros, têm permitido rápido desenvolvimento e crescimento da aplicabilidade de drones (BEKMEZCI; SAHINGOZ; TEMEL, 2013), o que insere novos requisitos nas redes de sensores, impactando diretamente na necessidade de mecanismos de sincronismo com maior precisão.

Os drones foram desenvolvidos com o objetivo de uso em aplicações militares, entretanto o seu uso em aplicações civis tem crescido rapidamente nos últimos anos (SANNA; PRALIO, 2005). Entre as diversas aplicações civis e militares a utilização de drones para formação de infraestrutura de comunicação de apoio para missões de resgate ou missões militares vem ganhando destaque, uma vez que é possível montar, alterar e desmontar uma rede de comunicações de forma dinâmica, atendendo aos objetivos de cada missão. O trabalho realizado por (ZENG; ZHANG; LIM, 2016) analisa as características de uma rede de comunicação de dados com este objetivo, destacando inclusive a formação de uma rede com vários drones.

A utilização de uma composição de múltiplos drones para realizar missões tanto civis como militares permite a captura de dados em modo paralelo o que impacta na redução do tempo de missão, no aumento da disponibilidade do sistema - isto é, a tarefa pode ser concluída mesmo se houver uma falha de hardware - e no aumento da capacidade de detecção destas aeronaves.

Este tipo de composição requer o estabelecimento de uma rede de comunicação de dados capaz de enviar informações para um ou mais nós e uma rede de comunicação de voz capaz de estabelecer comunicação entre o piloto remoto da aeronave, localizado em uma estação de controle em solo e o controlador de tráfego aéreo quando esta aeronave estiver inserida dentro de um contexto de espaço aéreo controlado. Este último caso normalmente ocorre em aplicações civis e ainda depende de regulamentação pelos órgãos responsáveis (EUROCONTROL, FAA, DECEA).

Durante a execução de uma missão podem existir situações inesperadas que alterem as características da composição dos drones, tais como: condições climáticas, interferências eletromagnéticas, perda de uma aeronave, falha de hardware e até mesmo as características geográficas da área podem gerar uma perda de comunicação entre GCS e a aeronave. Portanto, a FANET deve ser capaz de permitir

uma rápida recomposição para manter a capacidade de comunicação.

Para todas as aplicações citadas acima os protocolos de comunicação, de roteamento e de sincronismo utilizados em outros tipos de aplicações móveis precisam ser adaptados e melhorados.

A fim de manter a integridade da informação capturada e enviada, a integridade da comunicação de voz (onde aplicável), a agregação e recomposição correta da informação detectada (drones podem capturar partes de informação e, em seguida, enviá-las para a GCS a qual executa a agregação e consolidação das informações recebidas) e o estabelecimento de localização geográfica precisa da aeronave, é necessário um mecanismo de sincronização de alta precisão. Este mecanismo deve ser hierárquico e deve possuir a capacidade de se adaptar ao longo da missão, se necessário.

Para estes casos a precisão de um receptor de GPS pode não ser suficiente. A estimativa do erro de posição definido como a distância entre a posição real e o estimado pode variar de zero até algumas centenas de metros dependendo do fabricante do receptor GPS (ALEKSANDER et al., 2010). Esta precisão normalmente é aceitável para aplicações em MANETS, porém para as *Vehicular Ad Hoc Networks* (VANETs) a precisão do GPS é de aproximadamente 10 metros, o que para uso de prevenção de colisão em aplicações de carros autônomos em estradas auto gerenciáveis torna-se inseguro. O mesmo aplica-se para o uso em FANETs que devido ao grau de mobilidade e a velocidade de deslocamento dos nós demandam mecanismos mais precisos de sincronismo (BEKMEZCI; SAHINGOZ; TEMEL, 2013).

Devido aos requisitos de precisão de tempo exigidos pelas novas demandas de aplicação, é factível concluir que, a capacidade de prever os limites para a precisão de sincronização, considerando mobilidade, velocidade de deslocamento, cobertura e conectividade da rede é de relevante importância. No entanto, poucos métodos para tratar esta questão de forma conjunta têm sido explorados.

1.2 Objetivo

Diferentemente dos estudos citados, este trabalho tem como objetivo o desenvolvimento de um modelo de análise de sincronismo baseado em troca de mensagens que permita a avaliação da precisão de sincronismo da rede de sensores baseado nos requisitos da aplicação.

Este modelo de análise considera como variáveis a área de sensoriamento, a capacidade de transmissão e recepção dos nós sensores, a cobertura da área de sensoriamento, a probabilidade de perda de conectividade dos nós sensores distribuídos e a velocidade de deslocamento dos nós sensores.

O modelo desenvolvido considera o deslocamento dos nós em alta velocidade e baixa velocidade em um modelo de cobertura de área e manutenção da conectividade. Neste caso, para que um evento seja considerado detectado não é suficiente que ele ocorra dentro do raio de alcance de algum sensor, mas também é necessário que o sensor responsável por esta detecção possua conectividade com o nó sensor sink ou estação de processamento. Entende-se conectividade como a capacidade de um nó sensor qualquer transmitir e/ou receber mensagens de um nó sensor sink de forma direta ou através de outros nós sensores, independente da quantidade de saltos.

Dependendo do estado que os nós sensores estão, da mobilidade e do nível de energia, o sincronismo em um determinado nó pode ser diferente em determinados instantes de tempo.

Esta diferença de precisão de sincronismo da rede em função dos estados dos nós é a situação a ser medida e analisada, de forma que seja estabelecida uma relação entre a taxa de transmissão de mensagens de sincronismo, a quantidade de nós que estão no estado *habilitado*, a mobilidade e o nível de sincronismo obtido.

A partir do modelo desenvolvido para este estudo e dos resultados obtidos será possível projetar redes de sensores que atendam diferentes graus de sincronismo demandados por diferentes tipos de aplicação, em função da quantidade total de nós sensores, quantidade de nós sensores ativos, mobilidade e conectividade.

Como possíveis aplicações deste trabalho podem-se citar uma rede de sensores formada por drones, uma rede navios com sistema de rastreamento ou uma rede de

sensores para monitoramento climático. Considerando as FANETs, o estudo proposto permitirá que este tipo de rede possa ser projetada de forma a se obter o mínimo de aeronaves necessárias para atendimento aos requisitos da aplicação.

É importante citar também que como será possível conhecer previamente qual o grau de sincronismo uma determinada rede fornece, tendo como base a quantidade de nós sensores e a sua mobilidade, este estudo poderá ser aplicado como parte de uma possível solução para a coexistência de drones e aeronaves tripuladas no mesmo espaço aéreo controlado.

Outra possível aplicação para este estudo é na área de sincronismo entre drones, quando os mesmos estabelecem um voo totalmente autônomo.

1.3 Motivação

As redes de sensores tem sido largamente estudadas sob diversos aspectos. Porém a mobilidade dos nós sensores traz novos desafios a este tipo de rede. Enquanto um sensor que monitore rebanho de gado se desloca a velocidades extremamente baixas, um sensor em uma aeronave remotamente pilotada, por exemplo o modelo X-45A construído pela BOEING, se deslocará a uma velocidade de aproximadamente 919 km/h (OFFICE OF THE SECRETARY OF DEFENSE, 2002).

Os drones estão se tornando cada vez mais acessíveis e presentes em aplicações civis e militares e, portanto, as redes de comunicação formadas por estes tipos de nós vêm sendo estudadas para se romper os desafios impostos por estas aplicações.

Devido ao avanço tecnológico e à queda constante dos custos de construção, manutenção e operação destes equipamentos o sensoriamento de grandes áreas pode ser obtido a partir da composição de diversos sensores. Sendo que neste caso um drone possui somente um tipo de sensor e assim o cenário completo é reestabelecido com a sobreposição de imagens e informações das mais diversas naturezas, tais como temperatura, relevo, umidade, deslocamento de terra. Este tipo de arquitetura faz com que o tamanho dos drones seja menor, facilitando assim sua construção, redução de custos, autonomia de voo e dificultando a capacidade de ser detectado em caso de aplicações militares. Porém impacta na necessidade de que a

rede estabelecida por estas aeronaves durante o sensoriamento, quando utilizadas, por exemplo para transporte em grandes centros, forneça um grau de sincronismo elevado, da ordem de nano-segundos (ns).

Estes equipamentos podem ser utilizados em diversas situações, tanto militar como civil e em ambas as situações as próprias aeronaves podem ser utilizadas como sensores que além de enviar a informação para um centro de controle que permita o estabelecimento de uma consciência situacional, podem trocar informação entre si para coordenar de forma colaborativa a evolução do voo minimizando a necessidade de intervenção de um controlador ou piloto remoto. Aplicações logísticas e de transporte de passageiros em centros urbanos se beneficiam deste tipo de cenário.

Atualmente as regulamentações definidas pela *Federal Aviation Administration* (FAA), existentes nos Estados Unidos, determinam que o drone pode realizar voos para aplicações civis em território americano, desde que a aeronave seja remotamente pilotada e permaneça dentro da linha de visada visual do piloto remoto (FEDERAL AVIATION ADMINISTRATION, 2016).

Para que possa haver uma expansão na área de cobertura de um voo de um drone, é necessário que o nível de confiabilidade das informações trocadas entre a aeronave e o piloto seja elevado e as capacidades do sistema como um todo também devem ser confiáveis.

Estas capacidades podem ser afetadas pela carga de informação sensoriada a ser transmitida, pela carga de informação de controle de voo da aeronave, por ausência de algum nó no momento da transmissão devido ao distanciamento entre os nós ou devido a interferências de diversas fontes, eletromagnéticas ou naturais, na comunicação.

Outras situações em que não existe a formação de uma FANET, porém é necessário o sincronismo, são as aplicações de controle de tráfego aéreo, detecção de alvo hostil, monitoramento de fronteira e sensoriamento em área de desastre. Este tipo de aplicação depende de uma rede de sensores que é formada por nós fixos (por exemplo radares), normalmente um nó móvel e uma estação de solo capaz de receber os dados dos diversos sensores e efetuar a fusão destes dados de forma a obter a informação completa, conforme apresentado, de forma simplificada, na Figura 2

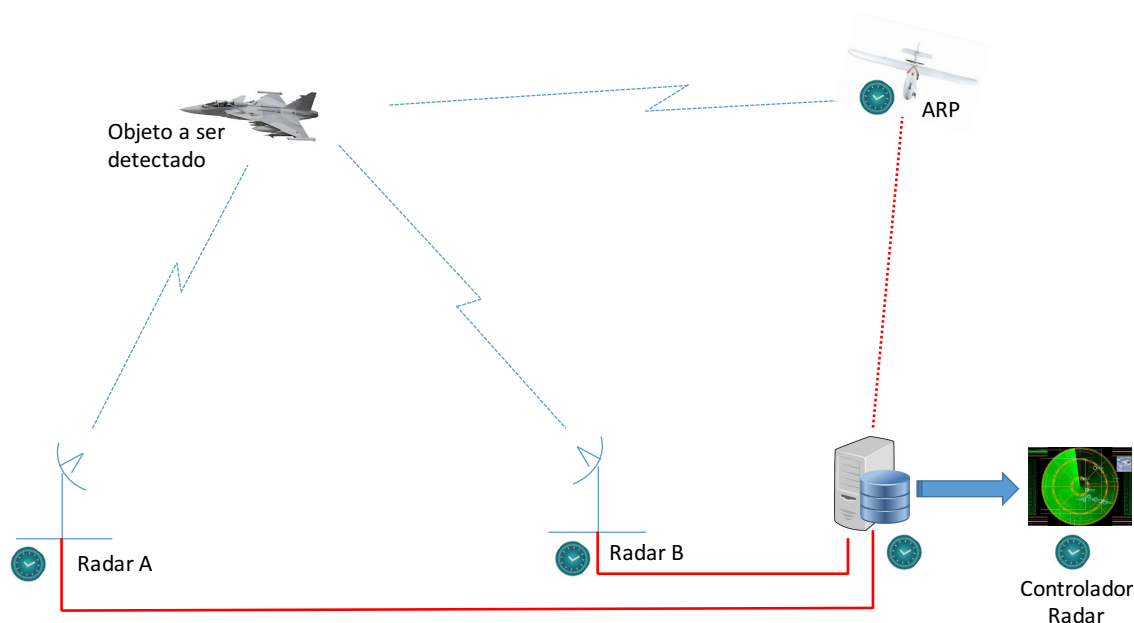


Figura 2 - Detecção Radar e Fusão de Dados

Fonte: Elaborada pelo Autor

Para o caso específico de controle de tráfego aéreo a estação de solo deve agregar os dados obtidos de diversos radares e apresentar a informação de posicionamento do objeto detectado. Para que todo o processo de agregação seja realizado de forma precisa, todos os sensores e a estação de solo devem possuir um sistema de referência horária baseada em GPS. E todos os nós sensores devem possuir um canal de comunicação ponto a ponto com a estação de solo, isto faz com que se tenha uma maior probabilidade de o atraso do canal de comunicação ser simétrico e constante.

O processo de detecção e agregação de dados ocorre da seguinte forma: quando um objeto é detectado por um sensor fixo, esta informação é enviada para a estação de solo. Como o sensor possui uma referência horária baseada em GPS, o pacote de dados a ser enviado recebe um *timestamp* com a informação do instante de tempo em que o objeto foi identificado.

A estação de solo ao receber o dado detectado pelo sensor, verifica o *timestamp* do pacote de dados e o compara com a sua referência horária, também obtida por GPS. Com esta comparação o sistema passa a conhecer o atraso do canal de comunicação e então é capaz de calcular o quanto o alvo se deslocou durante o tempo gasto na transmissão da informação. Quando existe mais de um sensor que detectou o mesmo objeto, o processo de cálculo de atraso do canal de comunicação é realizado

novamente. O ponto crítico deste método é que se qualquer um dos nós ou a estação de solo perder a referência GPS obrigatoriamente será inserido um grau de imprecisão no posicionamento do alvo detectado pois o atraso de transmissão será calculado com base em uma informação imprecisa.

De acordo com as recomendações vigentes o sistema de controle de tráfego aéreo tem como requisito mandatório uma média de erro quadrática de 300m e como requisito recomendado uma média de erro quadrática de 210m (EUROCONTROL, 2012). Considerando que o alvo detectado seja uma aeronave padrão detectada por apenas um sensor fixo e com velocidade típica de 850 Km/h, um erro de 1 segundo no sincronismo entre o sensor e a estação de solo, representará aproximadamente 236 metros de imprecisão.

Analisando o mesmo cenário descrito anteriormente, mas substituindo o sensor fixo por um sensor móvel (por exemplo um drone) tem-se no pior caso que a velocidade de deslocamento será a somatória da velocidade do alvo com a velocidade do nó sensor móvel (ambos os objetos se deslocando a velocidade máxima e em direção oposta). Considerando que um drone pode atingir a mesma velocidade de uma aeronave típica, tem-se como velocidade resultante 1700 Km/h, o que resultará em um erro de posicionamento de 472m para um erro de 1 segundo de sincronismo.

Apesar do erro apresentado romper o requisito para o sistema, a probabilidade de ocorrência de um desastre é baixa, pois atualmente as aeronaves tripuladas devem ter um espaçamento da ordem de 5.500 metros entre si. Em alguns casos esta distância de separação pode ser ainda menor (ICAO, 2007), porém o erro de posicionamento de um drone não deve interferir no espaço aéreo, uma vez que, atualmente, os drones não podem dividir o mesmo espaço aéreo (espaço aéreo controlado) utilizado pelas aeronaves tripuladas.

É importante destacar que o controle de tráfego aéreo não é considerado uma FANET, pois os enlaces de comunicação e troca de mensagens ocorrem somente entre aeronave e radares de detecção.

Porém ao extrapolar este cenário para a utilização de drones em locais densamente habitados, em aplicações de transporte de passageiros ou cargas, onde o voo ocorre entre os prédios dos centros urbanos, um erro da ordem de 236 metros pode ser considerado como um erro fatal. Em um cenário militar este erro representa atingir um

alvo não hostil, por exemplo um hospital ao invés de uma base inimiga. No caso de cenários civis, isto pode representar um choque entre um drone e um prédio, ou até mesmo com outro drone ou um helicóptero que esteja próximo a um ponto de pouso, o que inviabiliza aplicações logísticas e de transporte de passageiros, principalmente se analisarmos sobre o aspecto da realização de um voo completamente autônomo.

Há ainda que se considerar que:

- 1- Em um ambiente composto por diversos drones, não é possível estabelecer canais de comunicação ponto a ponto dedicados e com atraso simétrico e constante.
- 2- Em grandes centros urbanos, a utilização de drones para as diversas aplicações irá gerar um adensamento na ocupação do espaço aéreo e, portanto, a precisão de localização se tornará ainda mais crítica.

Portanto é fato que os métodos atuais não atendem as demandas para as FANETs ou aplicações onde existam diversos nós móveis se deslocando a velocidades elevadas.

Justifica-se assim a necessidade do desenvolvimento de modelos que permitam analisar o sincronismo em redes de sensores considerando variáveis como área de sensoriamento, a capacidade de transmissão e recepção dos nós sensores, a cobertura da área de sensoriamento, a probabilidade de perda de conectividade dos nós sensores distribuídos e a velocidade de deslocamento dos nós sensores.

1.4 Metodologia

A primeira etapa para a realização do trabalho consiste em definir, com base nas normas de sincronização do *International Telecommunication Union - Telecommunication Standardization Sector* (ITU-T) e do *Institute of Electrical and Electronics Engineers* (IEEE), os aspectos essenciais para o estudo de sincronismo em redes de pacotes, bem como aspectos relacionados às redes de sensores móveis, tais como atraso no canal de comunicação, assimetria do canal, probabilidade de perda de mensagens, taxa de transmissão da mensagem de sincronismo, acesso ao meio, algoritmo de roteamento e comportamento do nó sensor.

Após a realização desta etapa serão definidas as variáveis significativas tendo em vista a construção de um modelo que permita quantificar o relacionamento destas variáveis.

Considerando que estas variáveis são quantidade de nós sensores habilitados, mobilidade e perda de comunicação devido a impossibilidade de alcance de transmissão é possível desenvolver um modelo baseado na transição de estados dos nós sensores que compõem a rede de sensores. Este modelo de transição de estados é baseado em Cadeias de Markov uma vez que a transição para um estado futuro depende somente do estado atual do nó sensor.

Para o estabelecimento do sincronismo demandado será utilizado o princípio estabelecido no protocolo IEEE 1588, que é um mecanismo de sincronismo com precisão em nano-segundos para redes de pacotes, baseado em troca de mensagens. Este protocolo foi desenvolvido para uso em redes de telecomunicações e é capaz de entregar precisão na ordem de nano-segundos em ambientes SDH. Sua implementação permite compensar os atrasos dos relógios, atrasos inseridos por dispositivos intermediários além de tratar situações onde a troca de mensagens ocorra por enlaces assimétricos. Especificamente em redes de sensores móveis as mensagens trocadas entre os nós sensores usualmente não se utilizam dos mesmos links, uma vez que a necessidade de economia de energia, a existência de interferência e a mobilidade fazem com que os algoritmos de roteamento frequentemente alterem as rotas existentes entre os nós sensores.

É desenvolvido também um simulador com base no *Simulink*® da MathWorks (MATHWORKS, 2011), onde será possível a avaliação de cenários com o objetivo de verificar o comportamento da rede sob os aspectos de precisão de sincronismo em função da quantidade total de nós distribuídos na área de interesse e quantidade de nós sensores habilitados, considerando variáveis como conectividade, mobilidade e demanda de sincronismo.

Utilizando o simulador desenvolvido com base no modelo de estado dos sensores, são obtidos parâmetros de erro e de estabilidade do sincronismo. Com base nas normas ITU-T (ITU-T G.810, 1996) estes parâmetros são o *Maximum Time Interval Error* (MTIE) e *Time Deviation* (TDEV).

O modelo proposto é utilizado para definir situações limites onde os resultados do simulador podem ser comparados e validados.

Com a validação do simulador são implementados cenários mais complexos, variando a mobilidade dos nós sensores, a demanda de sincronismo da rede e a quantidade de nós sensores habilitados com o objetivo de analisar a precisão do sincronismo em um determinado nó sensor.

Ressalta-se que os desenvolvimentos necessários foram realizados de forma que possam ser reproduzidos em sistemas de plataforma livre, com o objetivo de maximizar o acesso à pesquisa desenvolvida, bem como facilitar sua reprodução e integração como parte de outras pesquisas em cenários mais complexos.

Baseado nos objetivos e metodologia de desenvolvimento deste trabalho ressaltam-se alguns conhecimentos os quais são indispensáveis para a sua implementação e futuras evoluções: Sincronismo em redes de Pacotes, Modelos de Processamento Distribuído, Teorias de Análise e Desempenho, Teoria das Filas, Telecomunicações, Engenharia de Software, Interconexão de Redes, Redes de Sensores, Algoritmos de Roteamento em Redes de Sensores, Algoritmos de voo e Normatização para voo não tripulado.

2 REDES DE SENSORES

Com o maior acesso às tecnologias de sensoriamento diversas aplicações tem se beneficiado deste tipo de rede. Uma rede de sensores pode ser implementada em locais com características muito diferentes uns dos outros, dependendo do seu objetivo. Em alguns casos estes sensores podem estar espalhados em terrenos remotos, como por exemplo florestas e campos agrícolas, ou até mesmo em campos de batalha. Para aplicações civis, estes sensores podem estar instalados em locais mais amenos, como por exemplo uma universidade ou indústria. Mas em ambos os casos os sensores serão em grandes quantidades o que impossibilita o estabelecimento de comunicação baseada em fio e dificulta a troca ou reposição de baterias. Portanto quando uma rede de sensores é projetada tem-se como requisitos o baixo consumo de energia e a utilização de comunicações sem fio, além da conectividade e mobilidade. Sendo que estes dois últimos requisitos têm se tornado mais importantes uma vez que a utilização de veículos autônomos, terrestres ou aéreos, para diversos objetivos, tem se tornado mais viável. Neste caso o sensoriamento passa a ser imprescindível para o atingimento de requisitos de segurança e confiabilidade.

De uma forma geral uma rede de sensores pode ser sistemicamente dividida em três componentes: sistemas, protocolos de comunicação e serviços, conforme mostrado na Figura 3 (YICK; MUKHERJEE; GHOSAL, 2008). O primeiro componente é o sistema. Cada nó sensor é um sistema e possui características específicas tais como capacidade de memória, tempo de vida útil de bateria, tipo de sensoriamento e tecnologia utilizada. O segundo componente constitui-se de protocolos de comunicação que permitem que a comunicação entre os nós e a aplicação, entre um nó e a estação base e/ou entre nós ocorra. Neste componente encontram-se protocolos responsáveis pelo roteamento e pela sincronização da rede por exemplo. O último componente são os serviços que são desenvolvidos com o objetivo de entregar para a aplicação uma rede com o desempenho e eficiência esperados.

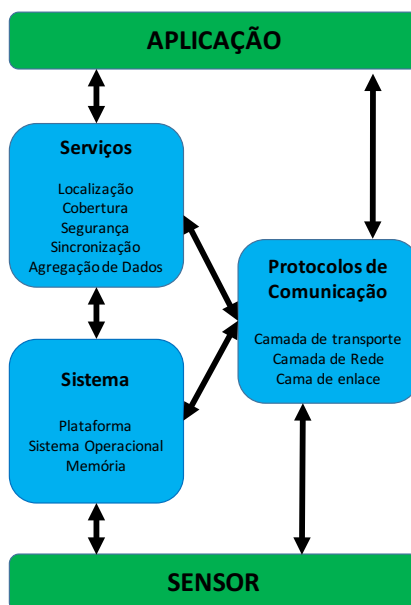


Figura 3 - Composição de um Rede de Sensores

Fonte: (YICK; MUKHERJEE; GHOSAL, 2008)

As características ou requisitos de uma rede de sensores são diretamente correlacionados com os requisitos da aplicação que demandará esta rede, e, portanto, podem variar dependendo do seu objetivo. Dentre algumas características pode-se destacar se a rede é *ad hoc* ou infraestruturada, se possui fontes extras de energia, como por exemplo painéis solares, se os sensores são moveis ou fixos, se a distribuição dos sensores é predeterminada ou completamente aleatória.

Considerando o aspecto da mobilidade, as redes de sensores passam a ter características específicas pois são influenciadas pela velocidade em que o sensor se desloca e podem sofrer interferências por obstáculos naturais, como por exemplo relevo, por obstáculos não naturais como prédios, interferências eletromagnéticas advindas de outros tipos de emissores tais como antenas de rádio e televisão e até mesmo radares utilizados tanto em ambientes civis para controle do tráfego aéreo como no ambiente militar para detecção de alvo ou aeronaves intrusas. Também podem sofrer interferência devido ao clima.

As redes de sensores móveis são chamadas de MANETS e possuem as seguintes características: rede sem a presença de infraestrutura, inexistência de um ponto central que controle a rede e a entrada e/ou saída de nós, sua arquitetura não é específica para alguma aplicação. Este tipo de rede foi desenvolvido para controlar e monitorar um grande espectro de eventos e aplicações.

Em aplicações em que os sensores são móveis, por exemplo drones utilizados para varredura de plantações ou drones utilizados para aplicações de transporte, é necessário compreender que os sensores possuirão duas funções bem específicas, e podem, inclusive, terem requisitos de confiabilidade, precisão, sincronismo e disponibilidades diferentes.

- Função 1: os sensores utilizados para coletar dados para aplicação fim. Isto é, em uma aplicação agrícola, os drones possuem sensores para coleta de imagem da plantação, temperatura e umidade entre outros.

- Função 2: os sensores são utilizados para a navegação dos drones. Neste caso os requisitos são mais rígidos pois a segurança do voo do drone depende da precisão dos dados de posicionamento.

Dentro do grupo das MANETS existem ainda as VANETS que são as redes veiculares e que podem ser consideradas um tipo particular de MANET. As VANETS podem ser basicamente de dois tipos Vehicle-to-Vehicle (V2V) e Vehicle-to-Road (V2R).

As redes V2V são as redes formadas entre veículos, onde a informação é trocada diretamente entre eles. Neste tipo de rede um determinado veículo pode por exemplo informar aos demais veículos que ele está desacelerando e, portanto, todos os demais veículos que receberem esta informação poderão de forma antecipada iniciar o processo de desaceleração, evitando assim a ocorrência de acidentes. Neste tipo de rede, a comunicação está limitada ao alcance de transmissão do rádio instalado no veículo.

As redes V2R são as redes formadas tanto pelos veículos como pela própria via onde estes veículos trafegam. Desta forma um veículo que detecte um acidente pode transmitir esta informação para todos os outros veículos através da comunicação com a via de tráfego, independente de estarem em seu raio de alcance ou não.

Características como alteração de topologia, padrão de mobilidade e velocidade de deslocamento são pontos importantes a serem considerados para as VANETs (CONTI; GIORDANO, 2014).

Seguindo a mesma premissa de segmentação, as redes de sensores formadas por drones são chamadas de FANETs. Estas redes possuem características específicas tais como, padrão de mobilidade, velocidade de deslocamento do nó, densidade da

rede, alteração de topologia, modelo de propagação do sinal de comunicação e poder computacional do nó (BEKMEZCI; SAHINGOZ; TEMEL, 2013).

2.1 FANET – Flying Ad hoc Networks

Uma das características mais importantes da rede de sensores que impacta diretamente na arquitetura, na eficiência e desempenho desta rede é a mobilidade. Apesar das FANETs, assim como as VANETs, serem consideradas em sua essência redes de sensores móveis, elas se diferenciam devido as características do ambiente em que são instaladas e principalmente devido aos padrões de mobilidade que cada uma destas redes possui.

As FANETs podem ser consideradas como um caso particular das MANETs e VANETs (BEKMEZCI; SAHINGOZ; TEMEL, 2013). Entretanto algumas características devem ser observadas:

- a) A velocidade de deslocamento dos nós em uma FANET é muito superior a velocidade de deslocamentos dos nós em uma MANET ou VANET. Para as FANETs os nós podem chegar a velocidades de 900 Km/h;
- b) Devido à velocidade de deslocamento dos nós a topologia é fortemente influenciada, podendo sofrer alterações constantes e com mais rapidez do que em outros ambientes;
- c) A comunicação em uma FANET pode ocorrer tanto ponto-a-ponto, ou seja, entre os drones para envio de informação ou estabelecimento de voo colaborativo, como também pode ocorrer entre drone e a Estação de Controle em Solo;
- d) Os enlaces de comunicação em uma FANET devem atingir distâncias maiores e estão sujeitos a intempéries naturais que afetam diretamente a qualidade do sinal e por consequência a capacidade de comunicação entre os drones e entre este tipo de aeronave e a Estação de Controle em Solo;
- e) Em uma FANET cada drone pode ser equipado com tipos diferentes de sensores. Por exemplo, um drone pode ser equipado com uma câmera do tipo *Forward Looking Infra Red* (FLIR) e outro drone pode ser equipado com um

sensor de detecção de alvo. Isto faz com que os requisitos de eficiência, desempenho, roteamento, transmissão de dados e atraso, entre outros, possa ser diferente para cada tipo de sensor. Neste tipo de situação pode ocorrer que cada drone execute uma parte do sensoriamento e posteriormente todas as informações serão agregadas para a formação do cenário completo, neste caso a precisão de sincronismo influencia diretamente na qualidade da informação obtida.

2.1.1 Pontos em Aberto

Apesar dos recentes avanços nos estudos das FANETs, algumas questões ainda necessitam ser aprofundadas e melhor discutidas. Entre elas é possível destacar as seguintes questões:

- a) Regulamentações – muitos países têm se empenhado para regulamentar os voos de drones. Porém as legislações atuais basicamente, exigem que seja solicitada uma autorização de voo aos órgãos competentes. No entanto, ao analisar as possibilidades do estabelecimento de uma rede com diversos drones e da realização de um voo colaborativo estabelecido entre estas aeronaves, as legislações carecem de aprofundamento. Por exemplo, a legislação brasileira, atualmente, não considera as aeronaves totalmente autônomas (DECEA, 2015). A coexistência de drones autônomos, remotamente pilotados, semiautônomos e aviões pilotados em um espaço aéreo controlado ainda é uma questão em aberto.
- b) Roteamento – o ambiente de comunicação em uma FANET pode mudar drasticamente em questões de segundos. Isto faz com que os protocolos de roteamento utilizados em uma FANET tenham que ser robustos o suficiente para se manterem atualizados e assim permitirem o estabelecimento das comunicações entre os nós. Segundo (SAHINGOZ, 2013) a maioria dos protocolos de roteamento utilizados nas MANETs vão falhar em um ambiente FANET.
- c) As rotas de sensoriamento estabelecidas para os voos das drones podem ser previamente estabelecidas. Porém em muitos casos estas rotas podem sofrer

- alterações em tempo real, como por exemplo em um ambiente militar, ou devido a uma intempérie natural. Desta forma é necessária a existência de algoritmos de planejamento que atendam à estas características e permita que as novas rotas possam ser recalculadas em tempo real.
- d) Requisitos de Qualidade de Serviço – QoS – as FANETS podem ter diferentes aplicações, desde aplicações militares até aplicações logísticas. Cada aplicação e cada tipo de sensor instalado no drone possui requisitos de largura de banda, atraso, latência, variação de atraso, precisão de sincronismo e taxa de perda de pacotes diferentes.
 - e) Colaboração entre aeronave tripulada e drones – o estabelecimento de voos híbridos, isto é, missões em que existam aeronaves tripuladas e drones remotamente pilotados ou autônomos é cada vez mais uma realidade. Para este tipo de situação deverá haver evolução nos sistemas de planejamento de missão, coordenação de voo, controle de espaço aéreo, enlaces de comunicação, sincronismo e legislações de forma que fiquem claramente estabelecidos os papéis e responsabilidades de cada tipo de aeronave.
 - f) A inexistência de regulamentação de frequências e protocolos de comunicações estabelecidas, isto faz com que haja forte interferência nos enlaces de comunicação quando ocorrem voos em grandes cidades, o que gera uma perda na qualidade do enlace de comunicação podendo em alguns casos acarretar uma perda total da comunicação. Este tipo de situação impacta diretamente na segurança do voo e por consequência pode causar danos à vida de pessoas.

3 SINCRONIZAÇÃO PRECISA DE TEMPO

Em geral, dois fenômenos são considerados sincronizados quando eles ocorrem simultaneamente. Porém quando se comparam dois relógios pode haver uma certa diferença entre eles. Esta diferença de tempo pode ser de dois tipos:

- a) A diferença entre os dois relógios é constante – neste caso, existe somente uma diferença de tempo ou de fase. Por exemplo, o relógio A mostra 15:00 hs enquanto o relógio B mostra 15:05 hs e, após um intervalo de tempo t , a hora apresentada pelos relógios A e B será $15:00 + t$ e $15:05 + t$, respectivamente;
- b) A diferença varia ao longo do tempo em si. Portanto, a diferença de tempo entre relógios A e B ao longo do tempo t , se altera. Por exemplo, às 15:00 hs a diferença entre os relógios é de 2 unidades de tempo, às $15:00 + t$ a diferença será de 3 unidades de tempo, às $15:00 + 3t$ a diferença será de 4 unidades de tempo, e assim por diante. Isto significa que a taxa de medição do tempo entre os dois relógios é diferente.

Quando existe uma diferença de taxa entre dois relógios, é implícito considerar que um deles é mais preciso que o outro e, portanto, o relógio menos preciso deve ser sincronizado considerando o relógio mais preciso como referência. Diferenças de tempo acima de um certo nível induz a erros em aplicações de sensoriamento e crescem a taxa de erro de bit (*bit error rate*) durante a transmissão dos dados.

Métodos analíticos para analisar o atraso têm sido propostos para uso em redes SDH. Entretanto estes métodos não podem ser aplicados em redes de pacotes devido a dificuldade na caracterização do atraso de trânsito neste tipo de rede.

3.1 Sincronização em uma Rede Wireless Mesh

As redes *Wireless Mesh* (*Wireless Mesh Networks – WMN*) envolvem comunicação direta entre as estações fim-a-fim, muitas vezes sem envolver uma estação fixa ou estação base.

Em uma WMN, os nós, após distribuídos, necessitam descobrir os seus nós vizinhos. Este conhecimento de sua vizinhança é essencial para quase todos os protocolos de roteamento, controle de acesso ao meio e algoritmos de controle da topologia (ASTUDILLO; KADOCH, 2014).

O padrão IEEE 802.11 (IEEE, 2012) define um modelo para cada tipo de ambiente. Estações próximas podem estabelecer relações e trocar dados entre si, porém devido ao movimento estabelecido pelas estações, estas relações passam a ser dinâmicas e conseqüentemente sofrem alteração ao longo do tempo.

Cada estação em uma WMN pode ser a fonte ou origem dos dados, o destino ou até mesmo ser um elemento de encaminhamento dos dados ao longo do caminho de comunicação estabelecido. A interconexão entre as estações da WMN e o mundo externo é realizada através de um elemento de rede denominado *mesh-gate*.

3.2 Sincronização em Redes de Pacotes

Considerando um cenário de rede wireless composto por nós sensores móveis, por exemplo drones, o *mesh-gate* recebe informação precisa de relógio de uma fonte externa e distribui esta informação para dentro do domínio da rede wireless.

O atraso para a transmissão desta informação é altamente variável, devido as rotas internas da rede se alterarem rapidamente, seja por motivo de mobilidade dos nós, ou por motivo de inoperância de um determinado nó. Além disso, o tempo de permanência da informação em uma estação pode ser diferente de outra estação ao longo do caminho de comunicação, devido a taxa de ocupação desta estação ou até mesmo da capacidade de processamento das estações.

O protocolo IEEE 1588v1 tem como premissa que o atraso na rede entre dos nós quaisquer, é simétrico, ou seja, é o mesmo atraso em ambas as direções. Esta hipótese é razoável para a maioria das tecnologias empregadas em redes cabeadas, mas não é aceitável para as redes *wireless mesh*. Neste caso, quando a estação A envia uma informação para a estação B, é estabelecido um caminho através da rede, porém quando a estação B enviar uma informação para a estação A, o caminho a ser estabelecido, frequentemente é diferente do caminho estabelecido anteriormente entre A e B, conseqüentemente o atraso do caminho entre A e B não será o mesmo do caminho estabelecido entre B e A. Esta diferença pode influenciar significativamente no estabelecimento de sincronismo de relógio das estações em métodos baseados em troca de mensagens, devido a não compensação de diferentes atrasos.

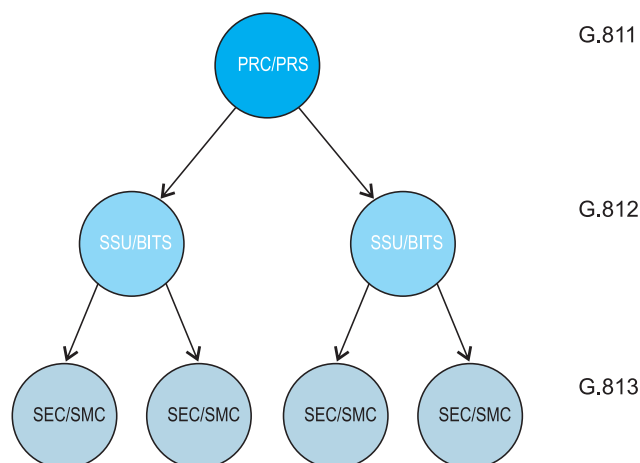
Devido a esta imprecisão que pode haver no sincronismo, o IEEE publicou uma nova versão do protocolo: o IEEE 1588v2 (IEEE, 2008) que adota mecanismos para lidar com atrasos assimétricos. Eles são chamados de relógio transparente. Este mecanismo foi desenvolvido para lidar com diferentes tamanhos de filas em switches de rede, mas a ideia pode ser adaptada para uma rede *wireless mesh*, uma vez que o objetivo é compensar atrasos em canais assimétricos e atrasos inseridos por tempo de processamento e encaminhamento das mensagens por equipamentos ou nós, no caso de redes de sensores, intermediários existentes na rede.

Uma possibilidade para o estabelecimento desta sincronização é a utilização do *Precision Time Protocol* (PTP) - definido no padrão IEEE 1588 - uma vez que a rede de comunicação de drones possui características semelhantes às de uma rede de pacotes.

Para as redes SDH o ITU-T padronizou uma hierarquia de sincronismo através das recomendações G.811 (ITU-T G.811.1, 2016), G.812 (ITU-T G.812, 2004) e G.813 (ITU-T G.813, 2003). Esta hierarquia tem sua referência primária em um relógio fundamental denominado *Primary Reference Clock* (PRC) a partir do qual toda a rede é sincronizada.

Um relógio de referência primário (PRC) gera um sinal de relógio com precisão da ordem de 10^{-11} equivalente em um relógio convencional a um segundo de erro em 3.172 anos. Os outros níveis possuem relógios de menor precisão intrínseca, mas cujo erro é limitado por uma comparação periódica com o PRC.

Os dois primeiros níveis da hierarquia de sincronização são geralmente utilizados com equipamentos específicos de geração de relógio, contendo osciladores de Césio ou de Rubídio. Já o terceiro nível é utilizado como parte de equipamentos de rede, tais como *switches* e roteadores. A Figura 4 apresenta esta hierarquia de sincronização.



Legenda:

PRC: Primary Reference Clock

PRS: Primary reference Source

SSU: Synchronization Supply Unit

BITS: Building Integrated Timing Supply

SEC: SDH Equipment Clock

SMC: Sonet Minimum Clock

Figura 4 - Hierarquia de Sincronização em Redes

Fonte: ITU-T G.811 (ITU-T G.811.1, 2016)

A troca de sinais de relógio entre diferentes nós, é implementada através da codificação da informação de tempo que é enviada pelo nó de referência.

Em redes de pacotes, os relógios de referência ainda são utilizados, mas a característica de transmissão assíncrona dos pacotes requer um novo método de distribuição de informação de tempo, pois esta informação deve estar incluída nos campos do pacote e o padrão IEEE 1588 (IEEE, 2008) trata justamente esta necessidade.

Para executar esta distribuição o padrão IEEE 1588 usa *timestamps* incluídas em pacotes de tempo especiais. Em redes de pacotes mecanismos de filas introduzem atrasos aleatórios e os pacotes de retorno podem ser enviados por diferentes caminhos, portanto a precisão deste método aplicado em redes de sensores depende

da frequência dos pacotes de tempo, da uniformidade do atraso, da utilização dos mecanismos de compensação definidos pela protocolo, bem como da conectividade e do grau de mobilidade dos nós pertencentes à rede.

3.2.1 Protocolo IEEE 1588

O padrão IEEE para a sincronização de tempo em redes de pacotes define um protocolo de sincronismo de duas vias. Esta denominação ocorre porque tanto o mestre (equipamento com o relógio mais preciso) envia pacotes eventos para o escravo (equipamento com relógio menos preciso) quanto o escravo envia pacotes eventos para o mestre. Porém, o fluxo de sincronismo é estabelecido sempre do mestre para o escravo.

Em 2008 foi definido o padrão IEEE 1588 v2 (IEEE, 2008) que tem como objetivo padronizar o mecanismo para a sincronização de relógios em sistemas distribuídos compostos de um ou mais nós e que estabelecem comunicação através de uma rede de pacotes. Este sincronismo é obtido através do *Precision Time Protocol* (PTP) o qual possui melhor precisão que outros protocolos como, por exemplo, o *Network Time Protocol* (NTP). Adicionalmente, suporta tolerância às falhas, às perdas de mensagens e ao recebimento de mensagens fora de ordem.

O funcionamento deste protocolo está baseado no conhecimento dos tempos de transmissão e recebimento de cada pacote de evento gerado pelos nós. O nó mestre observa o tempo t_1 , em que cada pacote evento é transmitido, em seu relógio local, convencionalmente mais preciso.

Esse valor deve ser enviado para o nó escravo no próprio pacote, ou, caso não seja possível deve ser enviado em um pacote subsequente. Quando o pacote é recebido pelo nó escravo, o instante de tempo t_2 em que o pacote chegou é observado. A diferença entre t_2 e t_1 pode ser calculada, dando origem a uma série com as diferenças de valores. Essas diferenças são filtradas e podem ser utilizadas para ajustar o relógio do nó escravo.

O processo se completa com o nó escravo enviando de volta para o nó mestre uma mensagem. Neste caso, o nó escravo mede e marca no pacote o instante de

transmissão da mensagem medido no seu próprio relógio enquanto que o mestre grava o instante em que o pacote é recebido. Estes instantes de tempo são denominados t_3 e t_4 , respectivamente.

A Figura 5 apresenta a troca de mensagens de sincronização estabelecida pelo PTP, no padrão IEEE 1588, entre o nó mestre e o nó escravo.

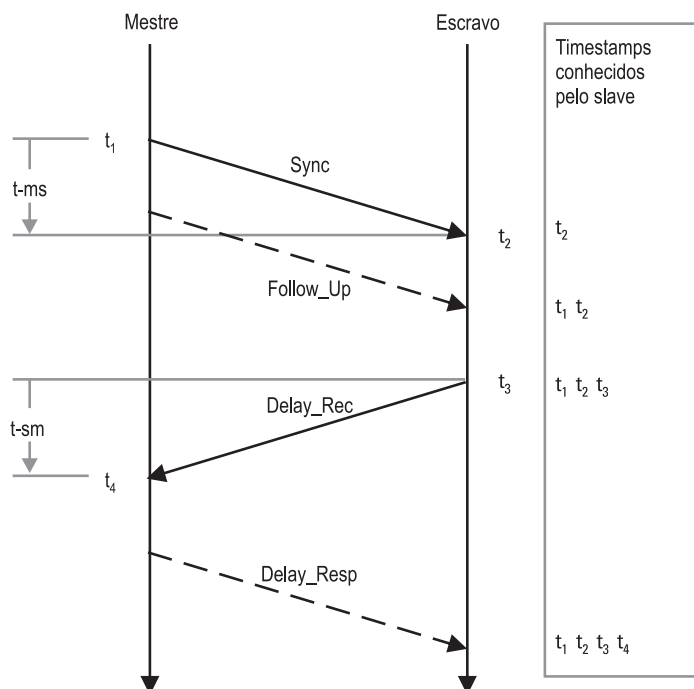


Figura 5 - IEEE 1588 - Mecanismo de Sincronismo

Fonte: IEEE 1588 (IEEE, 2008)

Uma vez conhecidos os tempos t_1 , t_2 , t_3 e t_4 , pode-se estabelecer o valor do *Offset* e do *Atraso*, os quais são obtidos a partir das equações (1) e (2), apresentadas a seguir:

$$t_2 - t_1 = \textit{Offset} + \textit{Delay} \quad (1)$$

$$t_4 - t_3 = \textit{Delay} - \textit{Offset} \quad (2)$$

Considerando $(t_2 - t_1) - (t_4 - t_3)$ obtém-se o valor do *Offset*, dado pela equação (3) apresentada a seguir:

$$Offset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \quad (3)$$

Considerando $(t_2 - t_1) + (t_4 - t_3)$ obtém-se o valor do *Delay*, dado pela equação (4) apresentada a seguir:

$$Delay = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \quad (4)$$

A principal simplificação deste modelo é considerar que o atraso nos dois sentidos é simétrico. Porém, em uma rede real, esta característica é improvável e o atraso pode variar de um momento para o outro em função do tráfego na rede, alterações de topologia e outros fatores. Isto pode gerar imprecisão nos tempos utilizados para correção. Esta imprecisão pode ser corrigida, se além do atraso médio, o valor do atraso em um dos sentidos também for conhecido.

A principal fonte de atraso variável ocorre no interior dos equipamentos de rede, como *switches* e roteadores, pelo fato do atraso de residência das mensagens depender das filas em cada sentido de transmissão. A compensação deste efeito pode ser obtida com um equipamento que mede este tempo de residência. O padrão IEEE 1588 especifica este equipamento como “relógio transparente”.

3.2.1.1 Extensão do Modelo para Atraso Assimétrico

A Figura 6 apresenta um exemplo da utilização do mecanismo do IEEE 1588 – relógio transparente – para compensar atrasos assimétricos. Para este exemplo foi admitido que existe um nó intermediário entre origem e destino e que os atrasos em cada direção são diferentes.

Os tempos, t_1 , t_2 , t_3 e t_4 são os mesmos eventos de tempo descritos anteriormente. E os nomes de mensagens utilizados, são os mesmos definidos pelo padrão IEEE 1588. É admitido que o nó de origem possui o relógio de referência e que existe uma diferença de *offset* em ambos os nós, intermediário e destino (*time sink*).

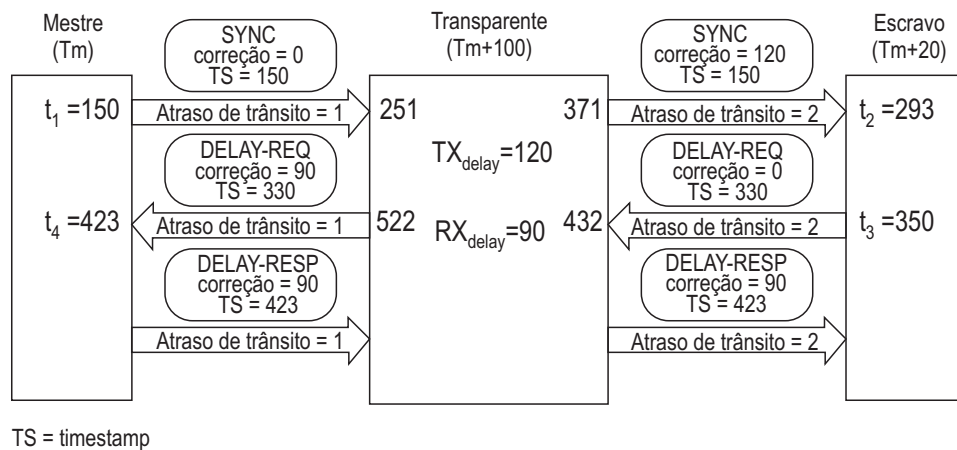


Figura 6 - Exemplo de Relógio Transparente com Atraso

Fonte: Elaborada pelo Autor

De acordo com o padrão a informação de correção está incluída em algumas mensagens para a contabilização do atraso assimétrico no nó intermédio. Estas correções são denominadas, TX_delay na direção de encaminhamento e RX_delay na direção oposta.

O dissipador de tempo pode calcular o atraso e o deslocamento utilizando as expressões (1), (2), (3) e (4) citadas anteriormente.

Desta forma, o valor do *Delay* e o valor do *Offset* são dados pelas equações (5) e (6) respectivamente e apresentadas a seguir:

$$Delay = \frac{(t_2 - t_1) + (t_4 - t_3) - TX_delay - RX_delay}{2} \quad (5)$$

$$Offset = (t_2 - t_1) - TX_delay - Delay \quad (6)$$

Usando as expressões 5 e 6 e aplicando-as no exemplo da Figura 6 o *Delay* (Atraso) e o *Offset* (deslocamento) podem ser facilmente calculados, resultando nos seguintes valores 3 e 20 respectivamente.

É importante ressaltar que com esta técnica a mensagem de atraso só inclui o atraso de trânsito, o que depende da propagação física no meio sem fios, mas não incluem os atrasos gerados internamente nas estações.

Os valores de TX_delay e $Delay$ são os valores do campo de correção da mensagem $Sync$ e o valor do campo de correção recebido na mensagem $Delay_resp$, respectivamente.

Este conceito de relógios transparentes resulta em um expressivo acréscimo na precisão obtida durante a sincronização. Este aumento de precisão levou o ITU-T a considerar o uso deste protocolo em aplicações de telecomunicações. Porém, quando as aplicações necessitam de precisão em fração de nanosegundo, mesmo o atraso diferencial em meios físicos pode ser significativo. Este atraso pode ser medido por um outro tipo de relógio denominado *Peer to Peer Transparent Clock*.

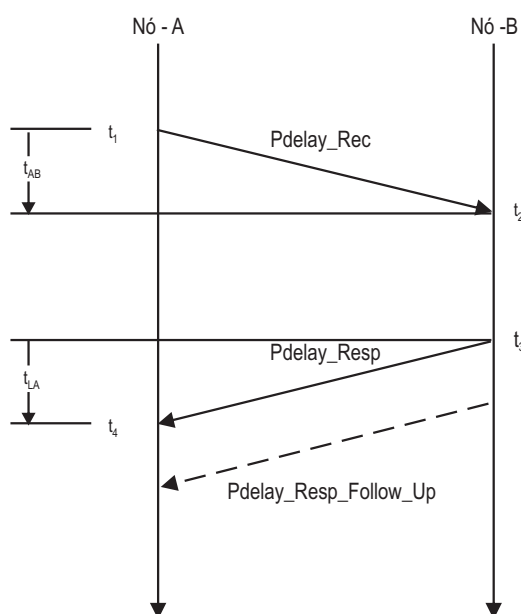


Figura 7 - Medição do Atraso Ponto-a-Ponto

Fonte: IEEE 1588 (IEEE, 2008)

O mecanismo de troca de mensagem executado pelo *Peer to Peer Transparent Clock* utilizado para a medição do atraso entre nós vizinhos é apresentado resumidamente na Figura 7.

O valor do atraso medido nos enlaces é acrescentado ao campo de correção das mensagens de forma a se obter uma melhora no efeito de compensação e consequentemente melhor precisão.

O uso de protótipos é limitado pelo custo de programas envolvidos. Existem modelos baseados no simulador OMNET++ (LIU; YANG, 2011) que tentam representar o

comportamento das estações envolvidas, mas o tempo de simulação cresce exponencialmente com as taxas de transmissão de rede.

Para o desenvolvimento deste trabalho, é proposta uma nova abordagem baseada na representação de eventos significativos da rede de sincronização. Na verdade, a precisão do sincronismo do relógio escravo depende da frequência da mensagem de tempo e valor de atraso da rede, mas não depende da forma específica de como esse conhecimento é obtido.

Desta forma, a abordagem proposta não representa o tráfego de dados da rede e pode ser aplicada mesmo em redes de alta velocidade.

3.3 Medição de Sincronismo

Para se obter o sincronismo de um ou mais relógios é necessário que exista um relógio de referência e o estabelecimento de um ponto inicial de contagem conhecido como época.

Uma vez que o relógio de referência é estabelecido, é possível verificar a existência de possíveis desvios de tempo entre um relógio qualquer e o relógio de referência e medir estes desvios. Isto significa que a medida do desvio ou erro de sincronismo entre dois relógios é sempre relativa.

A *International Telecommunication Union*, através do setor de padronização em telecomunicações, define na norma G.810 (ITU-T G.810, 1996) que matematicamente o erro entre um relógio qualquer e o relógio de referência no instante de tempo t é dado pela equação (7):

$$X(t) = T(t) - T_{\text{ref}}(t) \quad (7)$$

Em que:

$X(t)$ = Erro de tempo ou erro de sincronismo;

$T(t)$ = Tempo no relógio no instante de tempo t ;

$T_{ref}(t)$ = Tempo no relógio de referência no instante de tempo t .

A medição deste desvio pode resultar em valor positivo ou negativo. Quando o valor for negativo indica que um determinado relógio está atrasado em relação ao relógio de referência e quando for positivo indica que o relógio está adiantado em relação ao relógio de referência. A Figura 8 ilustra o deslocamento positivo e negativo do sinal de um relógio qualquer em relação ao relógio de referência.

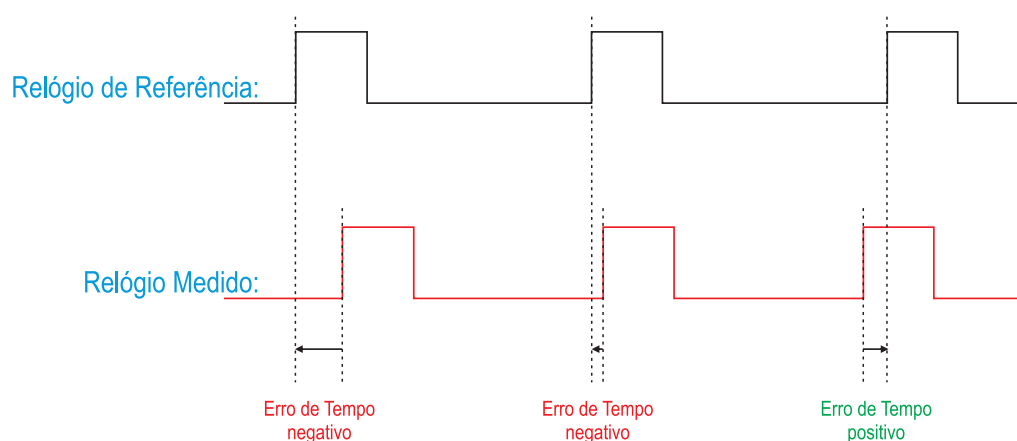


Figura 8 - Erro de Sincronismo

Fonte: Elaborada pelo Autor

A norma G.810 (ITU-T G.810, 1996) também estabelece cinco formas para avaliar a estabilidade de sincronismo de tempo, *Allan Deviation* (ADEV), *Modified Allan Deviation* (MDEV), *Time Deviation* (TDEV), *Root Mean Square of Time Interval Error* (TIErms), *Maximum Time Interval Error* (MTIE). A forma de cálculo de cada um destes parâmetros é apresentada a seguir.

3.3.1 *Allan Deviation* – ADEV e *Modified Allan Deviation* (MDEV)

O parâmetro *Allan Deviation* permite calcular as flutuações de frequência que ocorrem no domínio do tempo, em ondas não estacionárias, geradas tipicamente por osciladores de frequência tendo como base a variância entre duas amostras consecutivas.

Nos métodos estatísticos tradicionais, a variância trabalha com dados estacionários, na qual os resultados devem ser independentes do tempo. Já em osciladores, devido

às suas propriedades físicas, os dados de saída são não-estacionários, contendo componentes de ruído dependentes do tempo, o que afeta o valor do desvio de frequência (SIQUEIRA, 2014).

A média e a variância de dados estacionários tendem a convergir para valores específicos com o aumento do número de medições, porém para dados não-estacionários, sua média e a variância não convergem a um dado valor final. Assim, tem-se uma média móvel variando ao passo da inclusão de dados de novas medições, não havendo convergência para nenhum valor específico (KRONENBERG, 2007).

Desta forma, para a realização da análise da estabilidade de frequência no domínio do tempo, o método a ser utilizado é a variância de Allan ou mais comumente conhecido, o desvio de Allan que é obtida através da raiz quadrada da variância de Allan.

Devido a estas características o *Institute of Electrical and Electronics Engineers*, recomenda a utilização do cálculo do Desvio de Allan pelos fabricantes de geradores de frequência como padrão de especificação da estabilidade de frequência (KRONENBERG, 2007).

A característica que difere o MDEV do ADEV é que o MDEV é dependente do período da amostra (τ_0) enquanto que o ADEV não é dependente.

O cálculo de ADEV e MDEV são apresentados abaixo nas equações (8) e (9) respectivamente.

$$ADEV(\tau) = \sqrt{\frac{1}{2\tau^2} [(x(t+2\tau) - 2x(t+\tau) + x(t))^2]} \quad (8)$$

$$MDEV(\tau) = \sqrt{\frac{1}{2\tau^2} \left[\frac{1}{n} \sum_{i=1}^n (x_{i+2n} - 2x_{i+n} + x_i)^2 \right]} \quad (9)$$

Em que:

τ_0 = período da amostra;

τ = é o intervalo de observação dado por $\tau = \eta\tau_0$;

$x(t)$ = função de erro de tempo;

$\{x_i = (i\tau_0), i = 1, 2, \dots, N\}$ = sequência de N amostras igualmente espaçadas de $x(t)$.

3.3.2 *Time Deviation* (TDEV)

O parâmetro *Time Deviation* tem sido utilizado historicamente na caracterização do desempenho de sincronização, descrevendo a qualidade desta sincronização em redes de telecomunicações (DOBROGOWSKI; KASZNIA, 2010).

Diferentemente do ADEV e MDEV que são utilizados para medir a estabilidade de frequência de um oscilador, o TDEV é utilizado para caracterizar o erro de tempo de uma fonte de tempo (relógio) ou sistema de distribuição (RILEY, 1994). Sendo assim, é possível caracterizar a estabilidade de sincronismo existente entre dois relógios, assumindo que um deles seja o relógio de referência.

O parâmetro TDEV permite medir flutuações de fase inferior a 20Hz. Da mesma forma que o parâmetro MDEV o TDEV é dependente do período da amostra (τ_0).

O cálculo de TDEV é apresentado na equação (10).

$$TDEV(\eta\tau_0) = \sqrt{\frac{1}{6\eta^2} [[\sum_{i=1}^n (x_{i+2n} - 2x_{i+n} + x_i)]^2]} = \frac{\eta\tau_0}{\sqrt{3}} MDEV(\eta\tau_0) \quad (10)$$

Em que:

τ_0 = período da amostra;

τ = é o intervalo de observação dado por $\tau = \eta\tau_0$;

$x(t)$ = função de erro de tempo;

$\{x_i = (i\tau_0), i = 1, 2, \dots, N\}$ = sequência de N amostras igualmente espaçadas de $x(t)$.

O TDEV também pode ser obtido a partir do ADEV (PIETILÄINEN, 2008) e (RILEY, 2016), conforme a equações (11) e (12) abaixo:

$$\sigma_y^2(\tau) = \frac{1}{2(n-2)\tau^2} \sum_{i=1}^{n-2} (x_{i+2} - 2x_{i+1} - x_i)^2 \quad (11)$$

$$TDEV = \sigma_x(\tau) = \left(\frac{\tau}{\sqrt{3}}\right) \text{mod } \sigma_y(\tau) \quad (12)$$

Em que:

$\sigma_y^2(\tau)$ é a variância de Allan para diferença de fase.

3.4.3 Root Mean Square of Time Interval Error (TIErms)

Outro parâmetro constante da norma ITU-T G.810 é o TIErms. Assim como o parâmetro ADEV, o TIErms é independente do período da amostra. Para casos específicos onde não existe desvio de frequência, o TIE rms possui comportamento semelhante ao *Time Deviation*, porém o cálculo do TDEV permite identificar de forma mais adequada os tipos de ruídos presentes na rede de comunicações (RILEY, 1994).

O cálculo deste parâmetro é dado pela equação (13).

$$TIErms(\tau) = \sqrt{[x(t + \tau) - x(t)]^2} \quad (13)$$

Em que:

τ_0 = período da amostra;

τ = é o intervalo de observação dado por $\tau = \eta\tau_0$;

$x(t)$ = função de erro de tempo;

$\{x_i = (i\tau_0), i = 1, 2, \dots, N\}$ = sequência de N amostras igualmente espaçadas de $x(t)$.

3.3.3 Maximum Time Interval Error (MTIE)

O *Maximum Time Interval Error* é uma medida do erro de tempo máximo de um determinado relógio durante um intervalo de tempo específico. Basicamente o MTIE é calculado movendo uma janela ($\tau = \eta\tau_0$) através dos dados de fase (erro de tempo) e, a partir destas leituras, encontra-se a diferença entre os valores máximo e mínimo para cada posição desta janela. Devido a esta forma de cálculo o MTIE é muito sensível a um único valor extremo, mesmo que este valor seja transitório. Apesar desta sensibilidade o MTIE é muito utilizado na indústria de telecomunicações para se obter o erro de um determinado relógio (RILEY, 1994).

O MTIE é definido como a variação máxima de pico-a-pico de atraso de um sinal de temporização dada em relação ao sinal ideal num determinado intervalo de tempo ($\tau = \eta\tau_0$) para todos os tempos observados dentro do período de tempo medido (ITU-T G.8261, 2008), (ITU-T G.810, 1996). O tempo de observação é definido como ($\tau = \eta\tau_0$, sendo:

τ_0 = período da amostra;

τ = é o intervalo de observação dado por $\tau = \eta\tau_0$;

$x(t)$ = função de erro de tempo;

$\{x_i = (i\tau_0), i = 1, 2, \dots, N\}$ = sequência de N amostras igualmente espaçadas de $x(t)$.

A equação (14) apresenta a definição matemática para o MTIE (ITU-T G.8262, 2010)

$$MTIE(\tau) = \max_{1 \leq k \leq N-n} \left(\max_{k \leq i \leq k+n} x(i) - \min_{k \leq i \leq k+n} x(i) \right), \quad n = 1, 2, \dots, N-1 \quad (14)$$

3.3.4 Simulink® como Ferramenta para Aferição do Tempo

Esta seção apresenta algumas considerações no uso do *Simulink*® para análises de atraso e medidas nas diferenças de tempo.

O modelo representado na Figura 9 apresenta dois relógios configurados com geradores de pulsos e contadores. O número de pulsos contados é uma medida de

tempo em cada relógio de tal forma que o modelo calcula e apresenta a diferença entre os dois relógios.

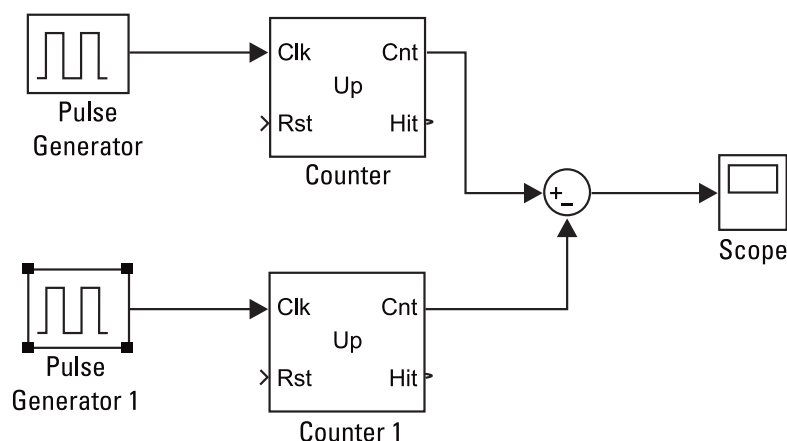


Figura 9 - Modelo Comparador de Pulsos

Fonte: Elaborada pelo Autor

Supondo que, inicialmente não exista diferença de frequência mas somente de tempo (ou de fase) dos geradores de pulso (o gerador 1 gera um pulso no instante 0,1 com período 1 e o gerador 2 gera um pulso no instante 0,3 com o mesmo período), o valor da diferença é sempre zero ou um, conforme apresentado na Figura 10.

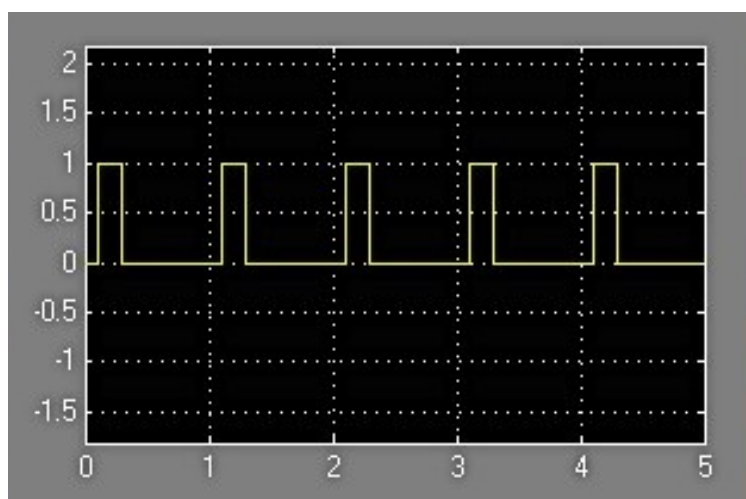


Figura 10 - Diferença de Tempo entre Geradores de Pulso

Fonte: Elaborada pelo Autor

Se houver uma diferença de frequência entre os dois relógios, a tendência da diferença é aumentar com o tempo. A Figura 11 demonstra um resultado intuitivo, isto é, se houver uma diferença de frequência, a diferença entre os relógios aumentará

indefinidamente, resultando na conveniência de sincronização entre os dois relógios após um determinado intervalo de tempo t .

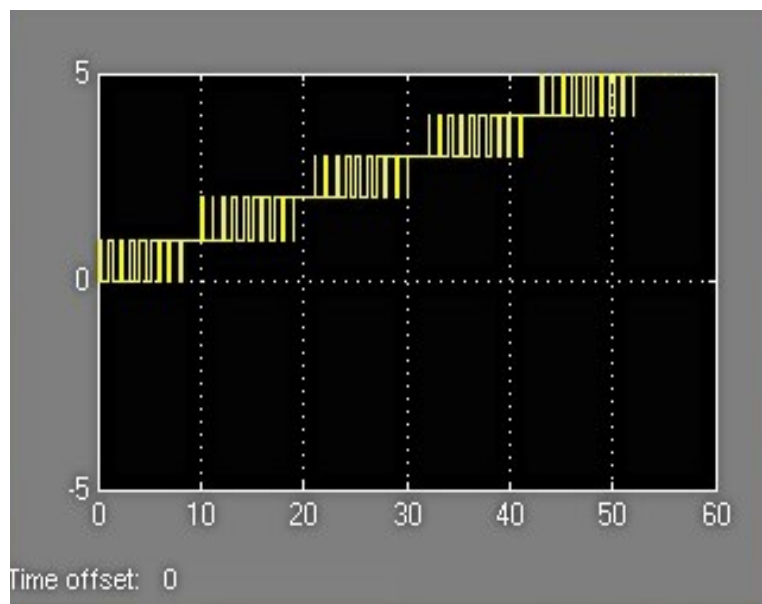


Figura 11 - Diferença de Frequência

Fonte: Elaborada pelo Autor

É claro que este problema não ocorreria se os dois relógios fossem igualmente precisos. Na prática, manter dois relógios (geograficamente distantes) sincronizados e precisos tem um custo elevado.

Para realizar o ajuste dos relógios de forma automática, sem necessidade de intervenção humana, pode ser utilizado um relógio cuja frequência do gerador de pulsos é ajustada. A Figura 12 apresenta um modelo em *Simulink*® com a característica de ajuste automático de frequência.

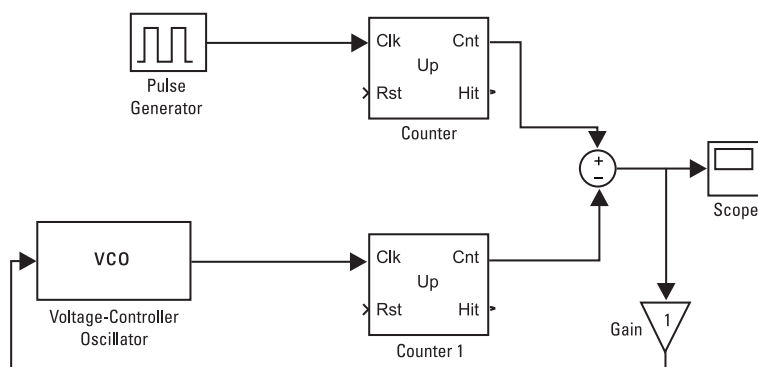


Figura 12 - Modelo com Ajuste Automático de Frequência

Fonte: Elaborada pelo Autor

A Figura 13 apresenta o resultado da simulação utilizando o modelo com ajuste automático de frequência. Apesar da diferença de frequência entre os dois relógios, a diferença de tempo se mantém limitada. Isto demonstra que é possível sincronizar relógios usando modelos de simulação. No entanto, sua aplicação ao ambiente de rede requer que o modelo proposto represente o sistema de comunicação entre as estações.

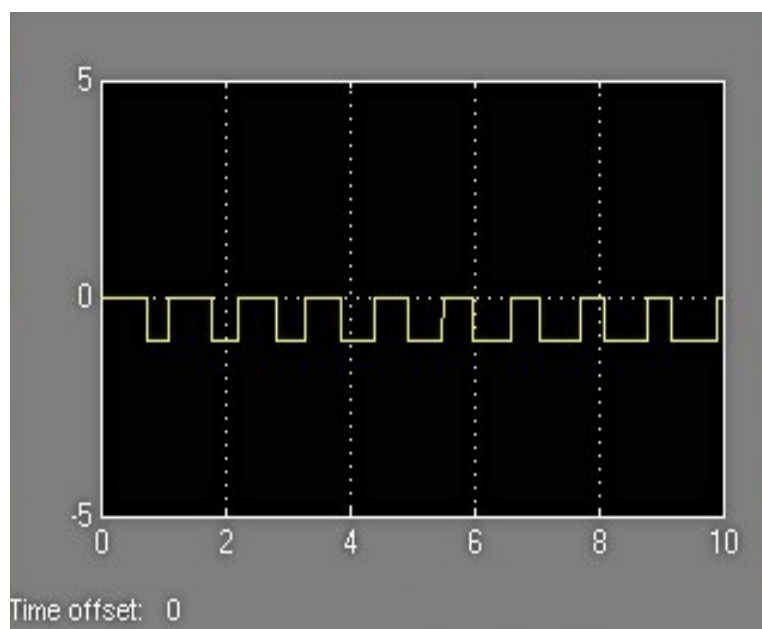


Figura 13 - Simulação com Ajuste Automático de Relógio

Fonte: Elaborada pelo Autor

Um modelo preciso de atraso de rede depende de diversos fatores e não é fácil construí-lo. Entretanto a análise do pior caso normalmente é suficiente para a validação de sua aplicação.

Para que um sistema realize o sincronismo entre dois ou mais relógios geograficamente distribuídos de forma automática é necessário que a diferença de tempo entre os relógios seja transmitida pela rede de comunicação. Isto pode ser feito de duas maneiras:

- Transmitir esta diferença de forma numérica, como ocorre nos padrões NTP (MARTIN; KASCH, 2010) e IEEE 1588 (IEEE, 2008);

- Transmitir o próprio sinal do gerador de pulsos, como ocorre nos padrões SDH e *Synchronous Ethernet* (SyncE) (ITU-T G.8262, 2010).

Em termos gerais, os sistemas se equivalem exceto pelas diferenças abaixo:

- a) Os padrões SDH e SyncE utilizam o sinal físico para a transmissão da frequência de pulso, portanto, são sempre sistemas ponto a ponto e não são afetados pelo tráfego da rede. O padrão SyncE pode ser estudado pela simulação de forma análoga ao SDH;
- b) Os padrões NTP e IEEE 1588 utilizam mensagens e podem transmiti-las entre pontos remotos da rede, porém, o tempo de transmissão entre estes pontos varia em função do tráfego na rede no momento em que ocorrer esta transmissão.

O receptor pode calcular o atraso da mensagem e o *offset* do relógio por meio das informações trocadas através do protocolo de mensagens, conforme explicado no Capítulo 3.

Estes princípios estão representados na Figura 14 e é utilizado como base para o desenvolvimento do modelo matemático e do simulador apresentado nos próximos capítulos deste trabalho.

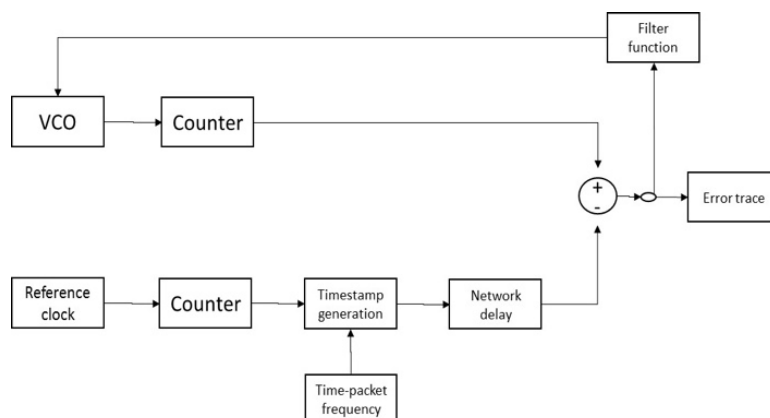


Figura 14 - Simulador Alto Nível

Fonte: Elaborada pelo Autor

4 DESENVOLVIMENTO DO MODELO

Este capítulo apresenta a modelagem matemática desenvolvida para a análise de sincronismo em redes de sensores, tendo como parâmetros: a quantidade de nós sensores, mobilidade, conectividade dos nós sensores e probabilidade de perda de comunicação.

Para o caso de mobilidade é apresentada a premissa adotada para caracterizar deslocamento dos nós sensores em alta velocidade e baixa velocidade, uma vez que esta caracterização impacta diretamente no estado da rede em um determinado instante de tempo t .

4.1 Visão Geral

Em uma rede de sensores móveis as vias de distribuição de sincronismo são variáveis, porque as relações de vizinhança são ajustadas dinamicamente de acordo com a distribuição espacial dos nós sensores.

Dependendo do grau de mobilidade dos nós sensores, esta distribuição pode ser mais intensa, impactando diretamente na probabilidade de conectividade de cada sensor. Por exemplo, em um ambiente FANET a velocidade de deslocamento dos nós sensores pode atingir 900 Km/h, desta forma, um drone pode, em um determinado instante de tempo, estar em uma posição onde não há rota viável até o relógio de referência – *Primary Reference Clock* (PRC) - impactando diretamente em seu sincronismo em relação ao nó sensor de referência.

Outro fator que influencia na capacidade de transmissão de mensagens é o estado que um nó sensor pode estar em um determinado instante de tempo, pois em redes de sensores o consumo de energia é um fator crítico e uma das formas de economizar energia é desligando o rádio do nó sensor por determinados intervalos de tempo.

Isto significa que, além da taxa de transmissão de mensagens do PRC, a mobilidade, a probabilidade de comunicação e a relação entre a quantidade total de nós sensores e quantidade de nós sensores habilitados influencia diretamente no grau de sincronismo da rede, pois as perdas de mensagens transmitidas são impactadas por todos estes parâmetros. Portanto no modelo desenvolvido, o sincronismo em um determinado nó sensor x_i é dado em função da relação de todos estes parâmetros. O diagrama em bloco da Figura 15 apresenta, em alto nível, o modelo desenvolvido.

O modelo implementa as características do padrão IEEE 1588 e combina uma representação realista do mecanismo de ajuste do relógio mais uma simulação funcional do comportamento da rede, a fim de obter simulações de duração aceitáveis.

A partir dos valores de sincronismo medidos em um determinado nó sensor x_i , em instantes de tempo diferentes, é possível estabelecer o valor de pico do sincronismo e parâmetros de estabilidade em um determinado intervalo de tempo. Estes valores definem o grau de sincronismo.

Com a implementação deste modelo será possível, estabelecer uma relação entre os parâmetros citados e o grau de sincronismo em uma rede de sensores.

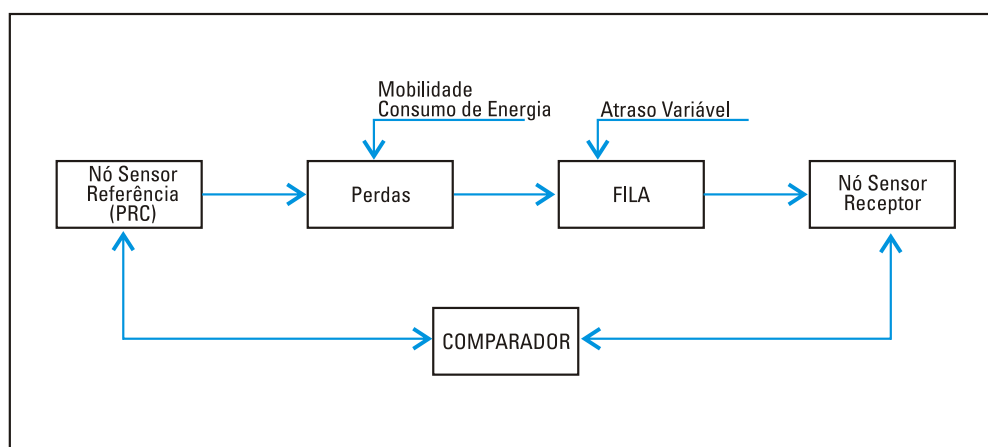


Figura 15 - Diagrama Alto nível

Fonte: Elaborada pelo Autor

4.2 Composição do Modelo

O modelo desenvolvido permite a simulação do comportamento dos elementos que compõem uma rede de sensores móveis, por exemplo uma FANET, sendo composto por uma quantidade de nós sensores distribuídos em uma área de sensoriamento circular de raio R . Cada nó sensor possui cobertura de transmissão circular de raio r , sendo que r é muito menor que R .

Nó sensor é definido como a composição de equipamentos, isto é, o rádio necessário para a transmissão e recepção de dados, equipamento de geolocalização, equipamento de mobilidade, por exemplo drone e o sensor propriamente dito, por exemplo um sensor de imagem ou sensor de temperatura.

Estes nós sensores assumem estados diferentes a cada instante de tempo devido a mobilidade ou consumo de energia dos mesmos.

Os elementos e a descrição do comportamento de cada um dos componentes são descritos a seguir.

4.2.1 Nós Sensores

Os nós sensores distribuídos dentro de uma área de sensoriamento podem ser de diferentes especificações. Por exemplo, podem existir sensores de temperatura, sensores de umidade, sensores de movimento entre outros. Nestas situações as informações coletadas são transmitidas de forma independente para uma estação de processamento centralizada e então são agregadas, compondo assim o sensoriamento completo da área. Em uma rede de sensores formada por drones, os sensores são instalados nos próprios drones e cada drone pode conter mais de um tipo de sensor. As informações coletadas por estes drones são transmitidas a um ponto central, onde ocorre o processamento e distribuição da informação agregada para outros centros de interesse.

Cada nó sensor pode estar *dentro do alcance* de transmissão ou *fora do alcance* de transmissão. Quando o nó sensor está *dentro do alcance* de transmissão, ele poderá assumir o comportamento de *habilitado* ou *não habilitado*.

O comportamento dos nós sensores quando estão nestas condições é dado pelos estados a seguir:

- a) *HABILITADO* – significa que o nó sensor está ativo e *dentro do alcance* de transmissão e, portanto, habilitado para: captação de imagens, sensoriamento, transmissão e recepção de sinais de controle, sincronismo e informações resultantes do sensoriamento realizado, tanto por ele mesmo quanto por outros nós sensores.
- b) *NÃO HABILITADO* – significa que o nó sensor não executa sensoriamento e não transmite mensagem resultantes de sensoriamento geradas por ele mesmo ou por outros nós sensores. Neste estado também não é possível receber, transmitir ou retransmitir mensagens de controle ou de sincronismo. Vale ressaltar que este é um estado previsto e que o nó sensor assumirá, com o objetivo de economizar energia.
- c) *FORA DO ALCANCE* – Caso um determinado nó sensor perca a comunicação com o nó sensor central PRC (*sink*), seja de forma direta ou através de outro nó sensor, o mesmo entrará em estado *FORA DO ALCANCE*. Neste estado o nó sensor permanece escutando o meio de forma a monitorá-lo com o objetivo de restabelecer a comunicação. Este estado pode ocorrer devido a mobilidade do nó sensor, que pode resultar em um posicionamento à uma distância maior do que o alcance de transmissão e recepção do seu rádio.

O alcance de transmissão de um nó sensor é dado por uma área circular de raio r , tendo como centro deste círculo o próprio nó sensor.

Uma vez que os nós sensores estejam *dentro do alcance* e *habilitados*, o modelo assume que todas as mensagens são transmitidas e recebidas, não considerando, portanto, perdas por ruídos, interferências ou qualidade do canal de transmissão.

Em relação ao algoritmo de roteamento, o mesmo é tratado de forma transparente para o modelo, isto é, quando os nós sensores estiverem no estado *habilitado*, assume-se que o algoritmo de roteamento é capaz de determinar uma rota válida para a entrega da mensagem.

Em relação à mobilidade dos nós sensores, considera-se que a mesma ocorra de forma aleatória e esteja limitada ao perímetro estabelecido pela área de sensoriamento. Desta forma a cada instante de tempo tem-se uma probabilidade de

um determinado nó sensor se mover e conseqüentemente estar dentro do alcance de transmissão e receber mensagens.

A distribuição inicial dos nós sensores se dá de forma aleatória dentro de uma área de sensoriamento pré-estabelecida.

4.2.2 Sincronismo

Para o estabelecimento do sincronismo dos nós sensores é utilizado o *Precision Time Protocol* (PTP), que é normatizado pelo IEEE através do padrão IEEE 1588.

O PTP é um protocolo que implementa mecanismos de compensação de atrasos nas transmissões, estabelece níveis de precisão da ordem de nanosegundos em sistemas distribuídos compostos por um ou mais nós e não requer a utilização de uma rede específica para este fim. Adicionalmente, suporta tolerância às falhas, às perdas de mensagens e ao recebimento de mensagens fora de ordem.

Estas características atendem diretamente aos requisitos de comunicação das redes de sensores, uma vez que estas redes são sistemas distribuídos compostos por diversos nós sensores, em alguns casos centenas de nós sensores, onde as comunicações são sem fio, o consumo de energia tem que ser baixo e que as mensagens podem chegar fora de ordem devido a características de arquitetura da rede e dos protocolos de roteamento. Ainda há que se considerar que toda a comunicação deve ser estabelecida em uma única rede, responsável por transmitir todo o tipo de mensagens, seja mensagem de controle, sincronismo ou mensagem de dados sensorizados.

Para a utilização do protocolo PTP no modelo considera-se que os nós sensores são distribuídos de forma aleatória dentro de um perímetro estabelecido, determinando assim a área de sensoriamento, e o relógio de referência (*Primary Reference Clock – PRC*) é um ponto fixo localizado no centro desta área.

Quando um determinado nó sensor recebe a mensagem de sincronismo o seu relógio interno é ajustado com base na informação recebida. Após este ajuste o relógio interno deste nó sensor se mantém atualizado com base em um oscilador interno até o recebimento de uma nova mensagem de sincronismo.

Em função dos estados que os nós sensores podem assumir em um determinado instante de tempo t , um determinado nó sensor pode demorar um intervalo de tempo maior que outro nó sensor qualquer para receber a mensagem de sincronismo, impactando diretamente no nível de precisão de sincronismo da rede.

4.3 Transição de Estados

A modelagem dos estados dos nós sensores é realizada através de um processo de Markov de tempo discreto, uma vez que a transição para o estado futuro depende apenas do estado atual. Isto significa que a cada instante de tempo o nó sensor pode ter seu estado atual alterado e neste instante esta informação é utilizada como dado de entrada no modelo. Do ponto de vista da mensagem de sincronismo os estados dos sensores apresentados na seção anterior (*FORA DO ALCANCE*, *HABILITADO*, *NÃO HABILITADO*), podem ser representados conforme a Figura 16.

É importante ressaltar que, se, no instante de tempo t_0 o nó sensor x_i estiver no estado *FORA DE ALCANCE* e no instante t_1 este mesmo nó sensor se mova em direção a outro nó sensor de forma a entrar no raio de alcance de transmissão, então, no instante t_1 , este nó x_i , obrigatoriamente estará no estado *HABILITADO* e poderá receber a mensagem de sincronismo.

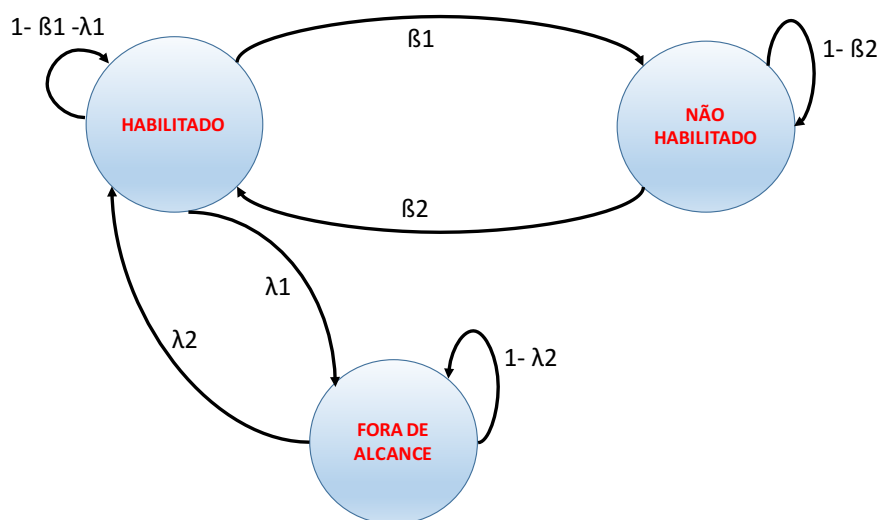


Figura 16 - Modelo de Transição de Estados

Fonte Elaborada pelo Autor

Os parâmetros β_1 e β_2 , representados na Figura 16 são as probabilidades de transição entre os estados *HABILITADO* e *NÃO HABILITADO*. Estas probabilidades modelam o consumo de energia, uma vez que a partir deste dado é possível estabelecer a relação entre a quantidade de nós sensores totais na rede e a quantidade de nós sensores que estão no estado *HABILITADO*. De uma forma simplificada pode-se afirmar que quanto maior o valor desta relação maior será o sincronismo da rede de sensores, uma vez que a quantidade de nós *HABILITADOS* e, portanto, com capacidade de transmissão e recepção das mensagens de sincronismo, será maior.

Os parâmetros λ_1 e λ_2 , representantes na Figura 16 são as probabilidades de transição entre os estados *FORA DE ALCANCE* e *HABILITADO*. Estas probabilidades modelam a mobilidade dos nós sensores na rede dentro da área de sensoriamento.

A mobilidade é modelada baseada na probabilidade de um nó sensor receber a mensagem de sincronismo e para que este recebimento ocorra o nó sensor deve estar no estado *HABILITADO*. Caso em um determinado instante de tempo o nó sensor, devido a mobilidade, esteja no estado *FORA DE ALCANCE*, então a mensagem de sincronismo transmitida não será recebida e conseqüentemente o sincronismo será afetado de forma negativa, isto é a diferença entre o relógio do nó sensor e o relógio do nó sensor de referência (PRC) aumentará.

4.4 Modelagem do Período de Repouso

Em redes de sensores o consumo de energia é um fator crítico que influencia diretamente na vida útil dos nós sensores. Uma das formas de economizar energia é alternar o estado dos nós sensores entre *HABILITADO* e *NÃO HABILITADO*. O período em que o nó sensor permanece no estado de *NÃO HABILITADO* é também chamado de período de repouso.

A modelagem do período de repouso é baseada na relação entre a quantidade de nós sensores no estado *HABILITADO* e a quantidade de nós sensores totais pertencentes a rede de sensores, distribuídos em uma área de sensoriamento (A_{SR}). Esta relação é denominada K_1 , e é estabelecida pelos parâmetros N_1 e N_2 , conforme equação 15.

Este parâmetro K_1 é a variável independente do modelo e será dado de entrada para a execução do mesmo.

$$K_1 = \frac{N_2}{N_1} \quad 0 < K_1 \leq 1 \quad (15)$$

Os parâmetros N_1 e N_2 são definidos da seguinte forma:

- N_1 é a quantidade total de nós sensores distribuídos de forma aleatória dentro do perímetro da área de sensoriamento.
- N_2 é a quantidade de nós sensores HABILITADOS dentro do perímetro da área de sensoriamento, portanto é um subconjunto de N_1 com a relação $N_2 \leq N_1$.

A Figura 17 ilustra a distribuição de nós sensores e seus estados em uma área de sensoriamento de raio R .

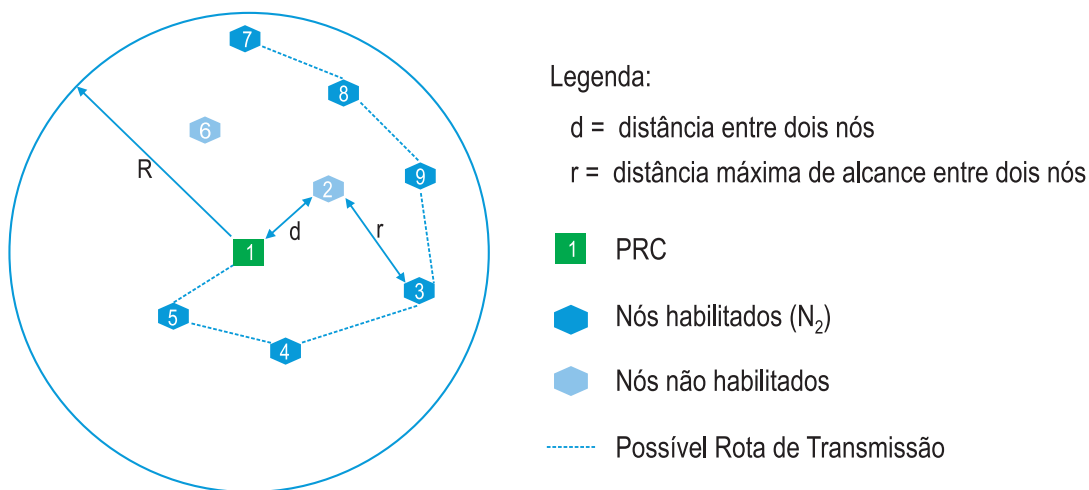


Figura 17 - Distribuição dos Nós em uma Área de Sensoriamento

Fonte: Elaborada pelo Autor

Quando o valor de K_1 aumenta, significa que a quantidade de nós sensores no estado *HABILITADO* é maior, portanto, a capacidade de transmissão de mensagens na rede também é maior, uma vez que a probabilidade de a mensagem de sincronismo ser transmitida para todos os nós sensores aumenta, resultando em um maior grau de sincronismo. Porém quando os nós sensores estão neste estado, o consumo de energia aumenta, o que significa que esta relação deve ser ajustada conforme a

necessidade demandada pela aplicação. Caso contrário a vida útil dos sensores, determinada pela capacidade de fornecimento de energia pela bateria, será reduzida. É esperado que a partir de um determinado valor de K_1 , o sincronismo atinja seu valor máximo, mesmo que nem todos os nós sensores estejam no estado *HABILITADO*. Isto ocorre porque não é necessário que todos os nós sensores estejam no estado *HABILITADO* para que exista uma rota válida para entrega de mensagens para todos os nós sensores existentes na rede, isto é, existe conectividade e, portanto, o sincronismo deixa de ser dependente dos demais parâmetros e passa a ser dependente somente da taxa de mensagens que o nó sensor central (PRC) envia.

4.5 Modelagem da Mobilidade

Para o tratamento deste parâmetro, foi implementado um componente no modelo, baseado na probabilidade, em um determinado instante de tempo t , de um determinado nó sensor estar conectado e, portanto, receber a mensagem de sincronismo originada pelo nó sensor de referência (PRC) dado que os nós sensores estão em movimento. Isto é possível pois esta probabilidade de comunicação é determinada em função da velocidade de deslocamento dos nós sensores.

A capacidade de entrega da mensagem de sincronismo na rede é dependente da existência de um caminho definido pelo protocolo de roteamento. Esta dependência é relacionada à distância entre os nós sensores e à velocidade de deslocamento destes mesmos nós sensores. Para a implementação do modelo adota-se como premissa que a mobilidade dos nós sensores sempre ocorre dentro de uma região de sensoriamento definida e de área conhecida.

A Figura 18 apresenta duas distribuições dos nós sensores em uma área de sensoriamento. No instante de tempo t_0 os nós sensores foram distribuídos de forma aleatória e, portanto, existe uma probabilidade de o nó sensor “3” ser capaz de receber mensagens, definida como $P_{t_0}^3$. No próximo instante de tempo t_1 , devido a mobilidade dos nós sensores, a distribuição destes mesmos nós sensores na área de sensoriamento será diferente e, portanto, existirá uma nova probabilidade de o nó sensor “3” ser capaz de receber mensagens, definida como $P_{t_1}^3$. Esta modelagem é

representada pelo parâmetro K_2 que determina exatamente a probabilidade de um determinado nó sensor receber a mensagem de sincronismo dado a sua mobilidade.

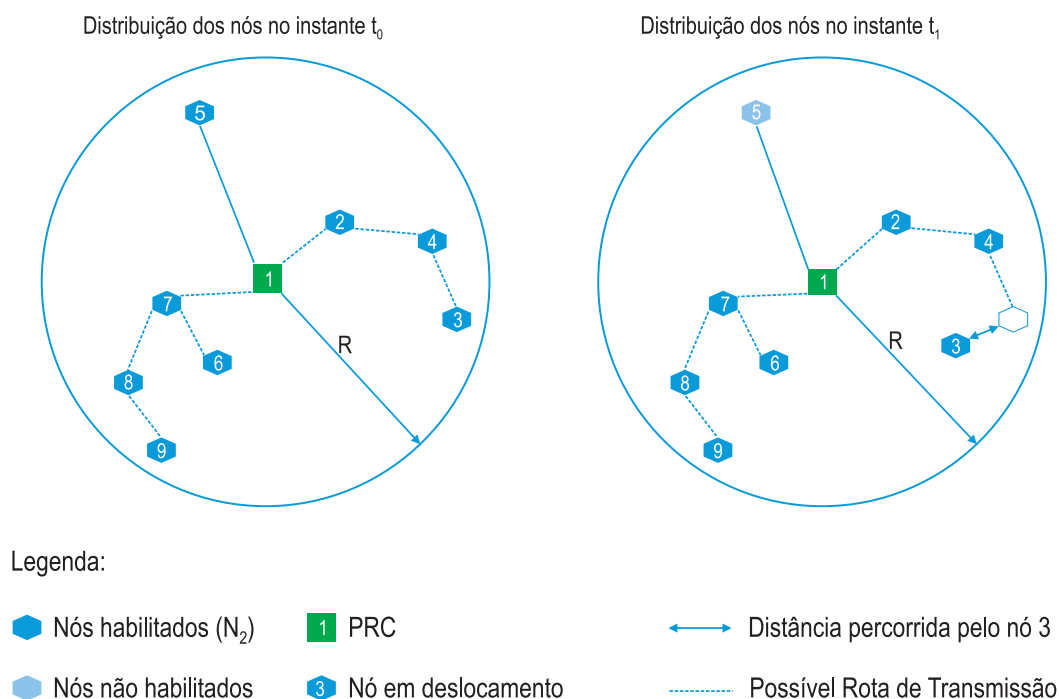


Figura 18 - Deslocamento do nó Sensor 3 no Instante t_0 e t_1

Fonte: Elaborada pelo Autor

Para a modelagem de K_2 , são utilizados os casos limites de mobilidade, isto é, velocidade elevada o suficiente para que a distribuição dos nós sensores seja completamente alterada no instante de tempo t_1 em relação ao instante de tempo t_0 e velocidade baixa o suficiente para que a distribuição dos nós sensores não perca a comunicação no instante de tempo t_1 em relação ao instante de tempo t_0 .

4.5.1 Caso 1 – Alta Velocidade de Deslocamento do Nó Sensor

Para determinar se a velocidade de deslocamento de um nó sensor, denominada V_d é alta, é realizada uma comparação entre a distância percorrida por este nó sensor, denominada D_p , e a distância de alcance de transmissão do rádio do nó sensor, denominada r , em um intervalo de tempo Δt (intervalo de simulação). A Figura 19

ilustra uma possível distribuição de nós sensores em uma área de sensoriamento e o alcance de transmissão dos rádios destes mesmos nós sensores.

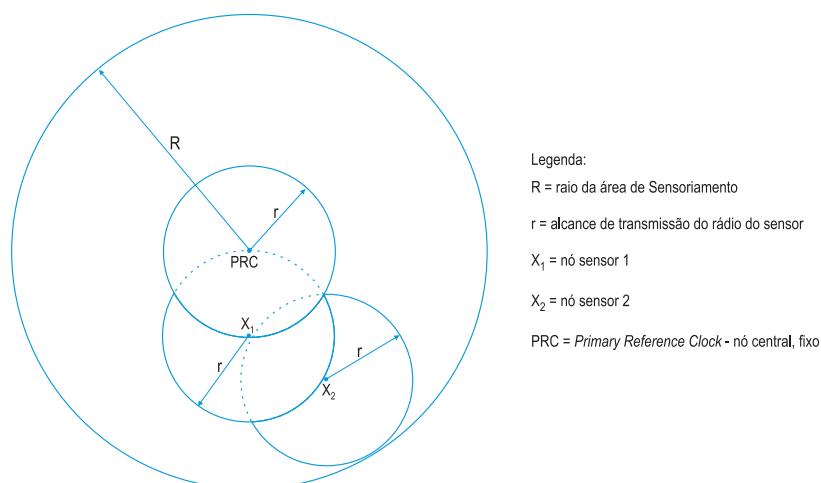


Figura 19 - Representação do Alcance de Transmissão dos Nós Sensores

Fonte: Elaborada pelo Autor

A distância percorrida pelo nó (D_p) é calculada conforme a equação (16):

$$D_p = \frac{V_p}{\Delta t} \quad \text{em que } \Delta t = (t_1 - t_0) \quad (16)$$

Se D_p é maior que r ($D_p > r$) então a velocidade de deslocamento é considerada alta e a probabilidade de um determinado nó sensor x_i receber a mensagem de sincronismo no instante de tempo t_0 é a mesma probabilidade deste mesmo nó sensor receber a mensagem no instante t_1 . Esta probabilidade é chamada de P_{random} conforme equação (17) abaixo:

$$P_{t_0}(x_i) = P_{t_1}(x_1) = P_{random} = P_{tk}(x_1) \quad (17)$$

Isto significa que a probabilidade de existir um caminho válido para o encaminhamento da mensagem até o nó sensor x_i em t_0 é a mesma em t_1 e, portanto, é como se em cada instante de tempo t_i os nós sensores fossem novamente distribuídos de forma aleatória na área definida.

4.5.2 Caso 2 – Baixa Velocidade de Deslocamento do Nó Sensor

Para que a velocidade de deslocamento do nó sensor seja considerada baixa é necessário que a distância percorrida por este nó sensor em Δt ($t_1 - t_0$) seja menor que a distância de alcance do rádio transmissor ($D_p < r$) deste mesmo nó sensor, considerando que em t_0 exista uma rota de comunicação válida entre o nó sensor e o PRC.

A probabilidade de o nó sensor x_1 receber a mensagem de sincronismo no instante de tempo t_k , admitindo que em t_0 esta mensagem foi recebida, é calculada com base na Figura 20.

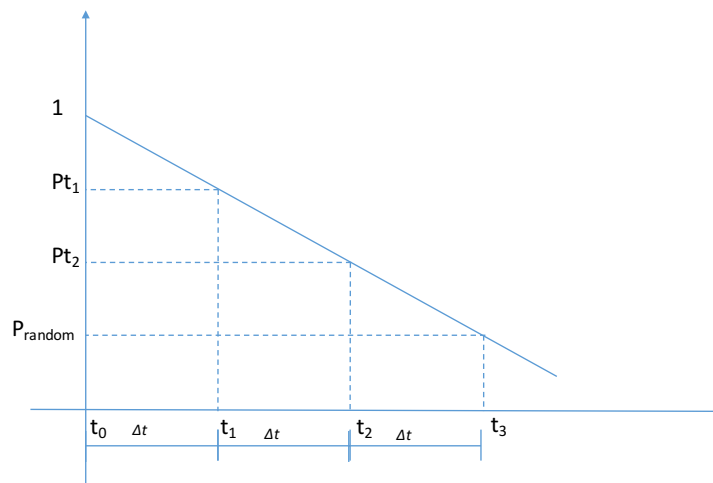


Figura 20 - Probabilidade de Recebimento de Mensagens

Fonte: Elaborada pelo Autor

Considerando que o número de passos necessários para que um determinado nó sensor perca a comunicação devido a distância percorrida seja dado pela equação (18).

$$n = \text{rounddown} \frac{r}{D_p} \quad (18)$$

e que o coeficiente de declividade (m) da reta é dado pela equação (19)

$$m = \frac{1 - P_{random}}{3 \Delta t} \quad (19)$$

e que Δt é o intervalo de tempo de integração, isto é $\Delta t = t_1 - t_0 = t_3 - t_2 = t_i - t_{i-1}$

então, temos que:

Para $n = 1$

$$P_{t_0}(x_1) = 1$$

$$P_{t_1}(x_1) = 1 - \frac{1 - P_{random}}{3 \Delta t} \cdot \Delta t$$

$$P_{t_1}(x_1) = \frac{2 + P_{random}}{3}$$

Para $n = 2$

$$P_{t_0}(x_1) = 1$$

$$P_{t_1}(x_1) = \frac{2 + P_{random}}{3 \Delta t} \cdot \Delta t$$

$$P_{t_2}(x_1) = 1 - \frac{1 - P_{random}}{3 \Delta t} \cdot 2 \Delta t$$

$$P_{t_2}(x_1) = \frac{1 + 2 \cdot P_{random}}{3}$$

$$P_{t_3}(x_1) = 1 - \frac{1 - P_{random}}{3 \Delta t} \cdot 3 \cdot \Delta t$$

$$P_{t_3}(x_1) = P_{random}$$

Com base no exemplo acima pode-se generalizar que a probabilidade de o nó sensor x_1 receber a mensagem de sincronismo no instante de tempo t_k é dada pela equação (20), sendo que $P_0(x_1) = 1$.

$$P_{t_k}(x_i) = 1 - \frac{1 - P_{random}}{n \Delta t} \cdot k \cdot \Delta t \quad 0 < K \leq n + 1 \quad (20)$$

4.5.3 Cálculo de P_{random}

As probabilidades de um nó sensor x_i receber a mensagem de sincronismo, tanto em alta velocidade quanto em baixa velocidade são dadas em função de P_{random} .

Para calcular P_{random} as seguintes hipóteses são adotadas:

- *Hipótese 1)* Dada uma área circular de raio R e nós sensores com alcance de transmissão r , existirão N_1 nós sensores distribuídos nesta área e N_2 nós sensores HABILITADOS, onde $N_2 \leq N_1$.
- *Hipótese 2)* Se a distância entre dois nós sensores é menor que r , a comunicação é possível. Caso contrário o nó sensor não recebe a mensagem.
- *Hipótese 3)* Os nós sensores se comunicam com um nó sensor central (S) que tem a função de PRC. Este nó sensor central está localizado no centro da área circular de raio R e é fixo. Este nó central existirá em todas as condições de simulação e modelagem e não está incluído no subconjunto de nós ativos (N_2).

Caso 1 – Dado que existe um nó sensor central (*sink*) com a função de PRC e um nó sensor x_1 é distribuído de forma aleatória em uma área de raio R , qual a probabilidade deste nó sensor x_1 estabelecer comunicação direta com o nó sensor central, se a posição deste nó sensor é determinada de forma aleatória? A Figura 21 representa a situação descrita.

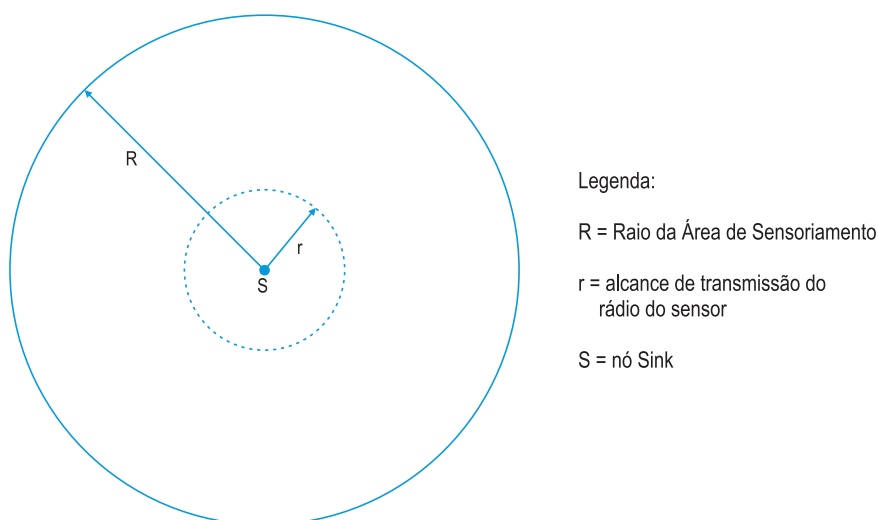


Figura 21 - Área de Alcance de Transmissão do Nó Sink

Fonte: Elaborada pelo Autor

Neste caso, a probabilidade de o nó sensor x_1 estabelecer a comunicação com o nó sensor central é a probabilidade de o nó sensor x_1 estar dentro da área de alcance de transmissão do nó sensor central de raio r , e é dada pela equação (21):

$$P(x_1) = \left(\frac{r}{R}\right)^2 \quad (21)$$

Caso 2 – Considerando 2 nós sensores HABILITADOS ($N_2 = 2$), distribuídos de forma aleatória dentro da área de raio R , qual a probabilidade de o nó sensor x_2 se comunicar com o nó sensor central (S) através do nó sensor x_1 ?

A Figura 22 apresenta a situação descrita. A ilustração (a) apresenta o alcance de transmissão a partir do nó sensor central (S) e do nó sensor x_1 .

Considerando que o nó sensor x_1 está a uma distância ρ do nó sensor central S, de tal forma que $0 \leq \rho \leq r$, um segundo nó sensor x_2 pode estabelecer comunicação com o nó sensor central através do nó sensor x_1 , se, e somente se o nó sensor x_2 estiver posicionado na área destacada na Figura 22 ilustração (b).

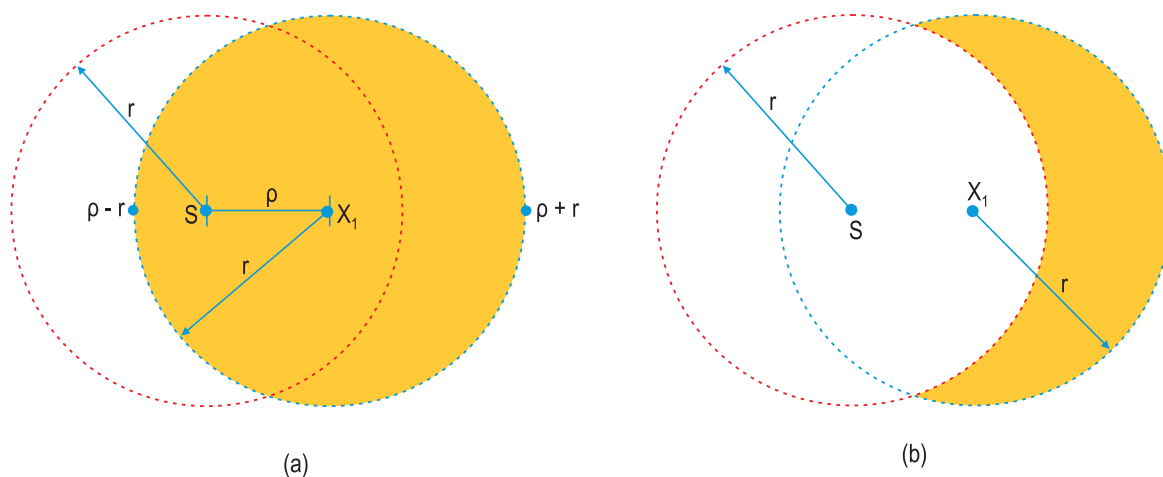


Figura 22 - Área de Alcance de Transmissão Total (Nó Sink + Nó X_1)

Fonte: Elaborada pelo Autor

Seja $D_s(x_k)$ a distância de um nó sensor K qualquer até o nó sensor central S, então para a situação apresentada na Figura 22 $D_s(x_1) = \rho$. Como ρ é uma variável aleatória contínua, a probabilidade de dois nós sensores, x_1 e S, terem D_s igual a zero, pode

ser determinada como nula, portanto, pode-se considerar D_S como uma pequena distância de largura $\partial\rho$.

As equações (22) e (23) apresentam a probabilidade do nó sensor x_2 se comunicar com S através do nó sensor x_1 .

$$P \{ \rho \leq D_s(x_1) \leq \rho + \partial\rho \wedge x_1 \text{ comunica com S} \} \quad (22)$$

$$P \{ \rho \leq D_s(x_1) \leq \rho + \partial\rho \mid x_1 \text{ comunica com S} \} \cdot P \{ x_2 \text{ comunicar com S} \} = \\ = \frac{2 \pi \rho \partial\rho}{\pi r^2} \cdot \frac{r^2}{R^2} = \frac{2 \rho \partial\rho}{R^2} \quad (23)$$

Seja E_p o evento cuja probabilidade foi calculada acima e apresentada na equação (23), $X_2 \rightarrow X_1$ o evento do nó sensor x_2 se comunicar com o nó sensor x_1 e $X_2 \rightarrow X_1 \rightarrow X_S$ o evento do nó sensor x_2 se comunicar com o nó sensor central S exclusivamente através do nó sensor x_1 , então $P \{ X_2 \rightarrow X_1 \rightarrow X_S \mid E_p \}$ é igual a probabilidade de o nó sensor x_2 estar na área destacada na Figura 22 ilustração (b).

O cálculo da área destacada é dado pela área do círculo formado pelo alcance do nó sensor x_1 que é igual a $A_{X_1} = \pi \cdot r^2$ menos a área estabelecida pela intersecção dos círculos formados pelo raio de alcance dos nós sensores, S e x_1 , denominada $A_{intersec}$.

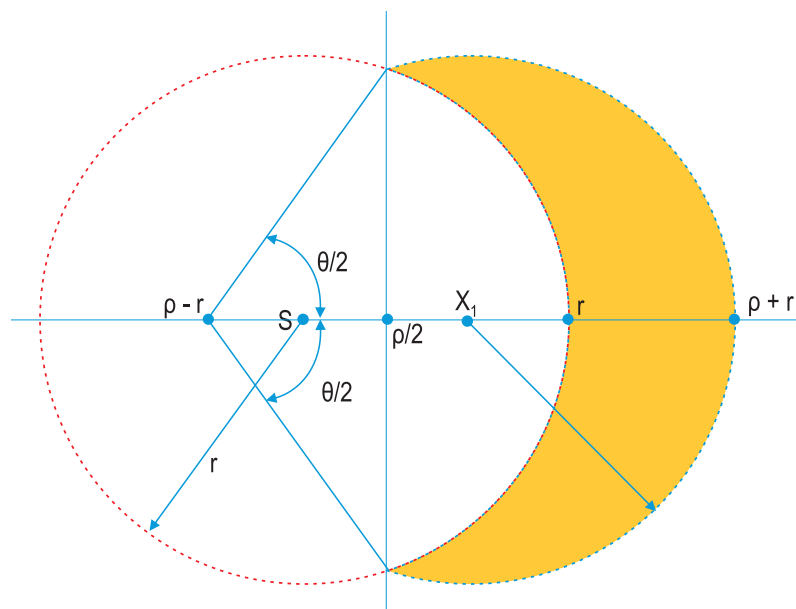


Figura 23 - Área adicionada pelo Nó X1 ao Alcance do Nó Sink

Fonte: Elaborada pelo Autor

A área destacada na Figura 23 é dada pela equação (24) abaixo.

$$A_{\text{destacada}} = \pi r^2 - r^2 (\theta - \text{sen } \theta) \quad (24)$$

Em que:

$$\theta = 2 \arccos (\rho / 2r) \quad \text{ou} \quad \cos (\theta/2) = (\rho / 2r)$$

A área de intersecção (A_{intersec}) é calculada conforme equação (25):

$$A_{\text{intersec}} = r^2 (\theta - \text{sen } \theta) \quad (25)$$

Para os casos limites, $\rho = 0$ e $\rho = r$, a área de intersecção terá os seguintes valores:

Para $\rho = 0$, então $\theta = \pi$ e, portanto, a área de intersecção será calculada conforme equação (26):

$$A_{\text{intersec}} = \pi r^2 \quad (26)$$

Para $\rho = r$, então $\theta = (2\pi / 3)$ e, portanto, a área de intersecção será calculada conforme equação (27):

$$A_{\text{intersec}} = 1,22 \cdot r^2 \quad (27)$$

Desta forma pode-se descondicionar a probabilidade apresentada na equação (23) de forma a eliminar o parâmetro ρ e então obter esta probabilidade em função dos parâmetros R (raio da área de sensoriamento) e r (raio de alcance de transmissão do nó sensor), resultando na equação (28), a qual é utilizada para a validação do modelo.

$$P\{X_2 \rightarrow X_1 \rightarrow X_S\} = \left(\frac{\pi r^4}{R^2} + \frac{4 r^4}{3 R^2} (\sqrt{3} - \pi) + \frac{r^4}{R^2} \left[\frac{16}{3} - 2\sqrt{3} \right] \right) \frac{1}{\pi R^2} \quad (28)$$

A resolução completa da equação 23 para se obter a equação 28 pode ser consultada no APÊNDICE A – RESOLUÇÃO EQUAÇÃO.

4.5.3.1 Generalização de P_{random}

O caso 2 apresentado e calculado na seção anterior tem como premissa um único nó sensor x_1 , posicionado dentro da área de alcance de transmissão do nó sensor central S , porém isto é apenas uma das possibilidades, uma vez que diversos nós sensores podem estar posicionados e HABILITADOS, em um determinado instante de tempo t , dentro da área de alcance de transmissão do nó sensor central S , dada por $\pi \cdot r^2$.

Neste caso o nó sensor x_2 , posicionado fora do alcance do nó sensor central S , conseguiria receber a mensagem enviada por S (PRC) através de um nó sensor x_i posicionado em qualquer lugar dentro da área de alcance de S .

Para a generalização do cálculo de P_{random} , são apresentados quatro casos de nós sensores posicionados dentro da área de alcance do nó sensor central S .

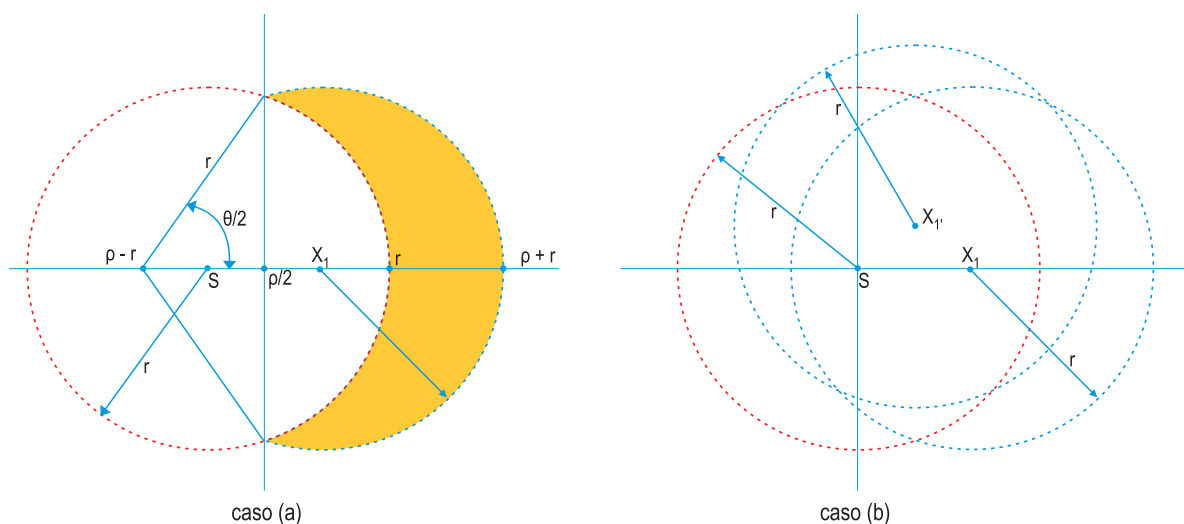


Figura 24 - Possível Posicionamento do nó X_1 - Casos (a) e (b)

Fonte: Elaborada pelo Autor

A Figura 24 apresenta dois dos quatro casos, são eles: o caso (a) onde a situação apresentada é a mesma discutida na seção anterior, e o caso (b) onde é apresentado um novo nó sensor posicionado dentro da área de alcance do nó sensor central S, denominado x_1 , que resulta em uma nova possibilidade.

Para ambos os casos (a) e (b), é necessário calcular a área acrescentada à área determinada pelo raio de alcance do nó sensor central S.

Para o caso (a), o cálculo da área total para alcance da transmissão é dado pela área do nó sensor central S, chamada de área fundamental, mais a área incorporada pelo raio de alcance do nó sensor x_1 . Portanto a área acrescentada é igual a área determinada pelo alcance do nó sensor x_1 , que é $A_{x_1} = \pi \cdot r^2$, menos a área de intersecção, ou área comum entre o nó sensor x_1 e o nó sensor central S, denominada $A_{intersec(X1/S)}$.

A $A_{intersec(X1/S)}$ em função da distância entre o nó sensor central S e o nó sensor x_1 (parâmetro ρ) é dada pela equação (29).

$$A_{intersec(X1/S)} = r^2 \left(2 \arccos \frac{\rho}{2r} - \text{sen} \left(2 \arccos \frac{\rho}{2r} \right) \right) \quad (29)$$

Portanto a área total de alcance será calculada conforme equação (30):

$$A_{\text{total de alcance}} = A_{\text{fundamental}} + A_{x_1} - A_{intersec(X1/S)} \quad (30)$$

Para o caso (b) a área de alcance acrescentada pelo nó sensor x_1 à área total de alcance considerando os nós sensores, S, x_1 e x_1' , é calculada pela equação (31):

$$A_{\text{acrescentada por } x_1'} = A_{x_1'} - A_{intersecção(X1'/S)} - A_{intersecção(X1'/X1)} + A_{intersecção(X1/X1'/S)} \quad (31)$$

O cálculo da área de intersecção de três círculos é uma questão já resolvida. A primeira etapa deve ser o cálculo do triângulo interno a área de intersecção. Este

triângulo é apresentado na Figura 25 e sua área é calculada pela fórmula de Heron, apresentada nas equações (32) e (33).

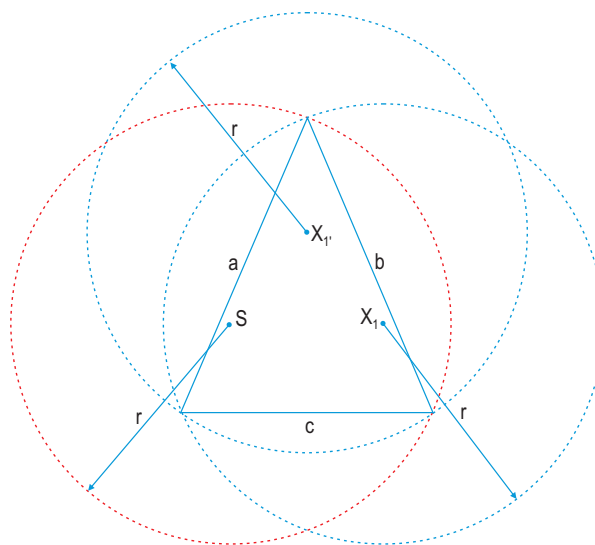


Figura 25 - Triângulo Formado pela Intersecção de Três Círculos

Fonte: Elaborada pelo Autor

$$A_{abc} = \sqrt{S(S-a)(S-b)(S-c)} \quad (32)$$

$$S = \frac{1}{2} (a + b + c) \quad (33)$$

As arestas a,b,c do triângulo podem ser calculadas com base nos pontos de interseção, a equação (34) apresenta a forma de cálculo para a aresta a, sendo que as arestas b e c são calculadas de forma análoga.

$$a = \sqrt{(X_{i1} - X_{i2})^2 + (Y_{i1} - Y_{i2})^2} \quad (34)$$

A área total de intersecção dos três círculos é dada pela equação (35).

$$A_{\text{intersecção}(X1/X1'/S)} = \sum_{n=1}^3 r_n^2 \text{sen}^{-1} \left(\frac{a_n}{2r_n} \right) - \sum_{n=1}^3 \frac{a_n}{4} \sqrt{4r_n^2 - a_n^2} + \sqrt{S(S-a)(S-b)(S-c)} \quad (35)$$

O terceiro caso, denominado caso (c) é apresentado na Figura 26. Para este caso um terceiro nó sensor, denominado $X_{1''}$ é acrescentado dentro da área de alcance do nó sensor central S.

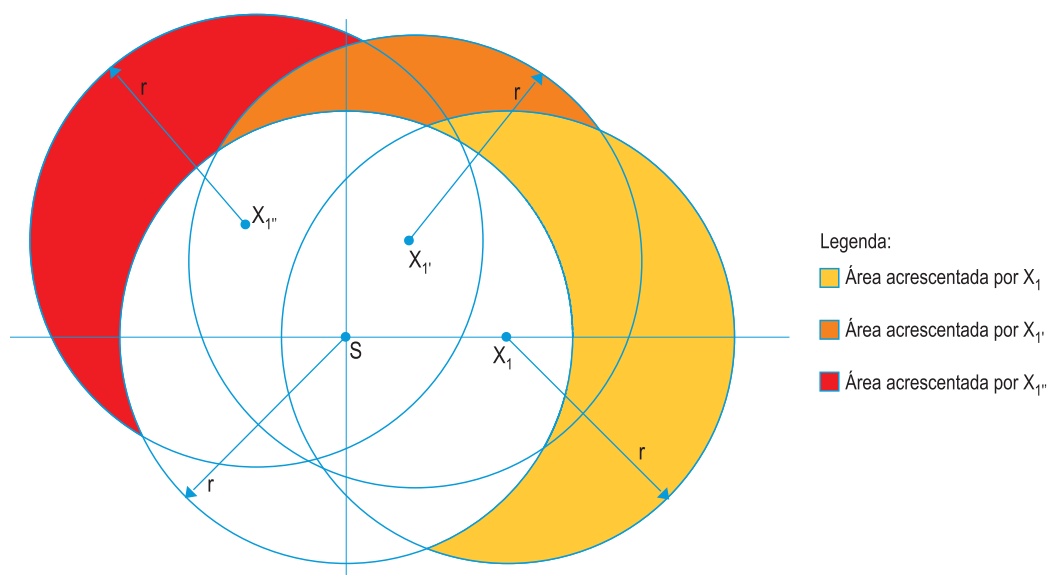


Figura 26 - Possível Posicionamento do nó X1 – Caso (c)

Fonte: Elaborada pelo Autor

Pode-se observar que a área acrescentada pelo nó sensor $x_{1''}$ não é dependente da área acrescentada pelo primeiro nó sensor, $x_{1'}$, e é determinada conforme a equação (36)

$$A_{\text{acrescentada por } X_{1''}} = A_{X_{1''}} - A_{\text{intersecção}(X_{1''}/S)} - A_{\text{intersecção}(X_{1''}/X_{1'})} + A_{\text{intersecção}(X_{1''}/X_{1'}/S)} \quad (36)$$

O quarto e último caso é apresentado na Figura 27, onde um quarto nó sensor, denominado $x_{1'''}$ é acrescentado dentro da área de alcance do nó sensor central S.

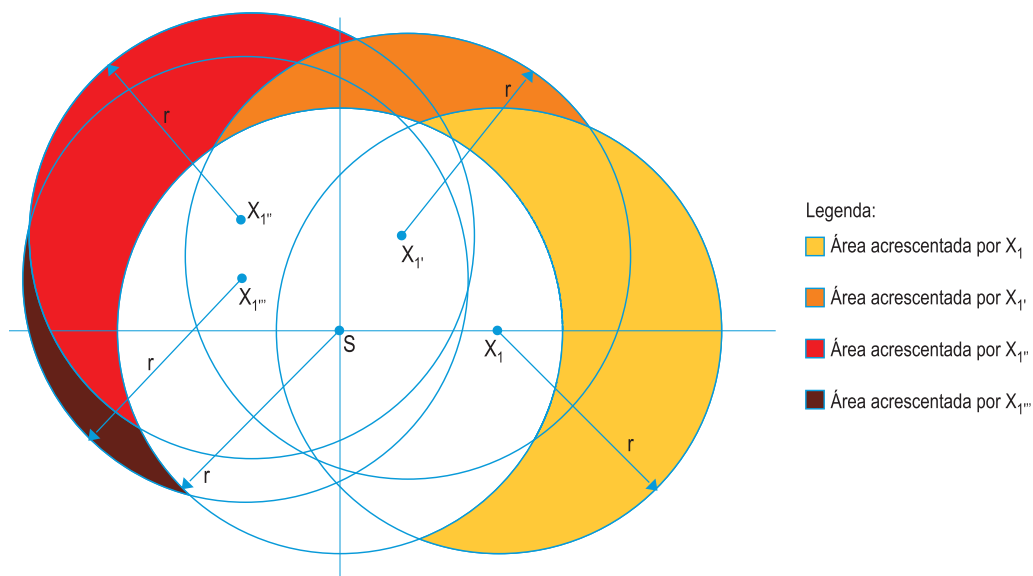


Figura 27 - Possível Posicionamento do nó X1 – Caso (d)

Fonte: Elaborada pelo Autor

De forma análoga aos outros casos apresentados anteriormente, a área acrescentada por este novo nó sensor é dada pela equação (37).

$$A_{\text{acrescentada por } X_{1'''}} = A_{X_{1'''}} - A_{\text{intersecção}(X_{1'''}/S)} - A_{\text{intersecção}(X_{1'''}/X_{1''})} + A_{\text{intersecção}(X_{1'''}/X_{1''}/S)} \quad (37)$$

A partir da análise dos quatro casos apresentados constata-se que a área acrescentada por um novo nó sensor qualquer x_{1i} , posicionado dentro da área de alcance do nó sensor central S, pode ser generalizada conforme a equação (38), apresentada abaixo, desde que todos os nós sensores x_{1i} sejam ordenados em forma crescente do ângulo α , conforme Figura 28.

$$A_{\text{acrescentada por } X_{1i}} = A_{X_{1i}} - A_{\text{intersecção}(X_{1i}/S)} - A_{\text{intersecção}(X_{1i}/X_{1i-1})} + A_{\text{intersecção}(X_{1i}/X_{1i-1}/S)} \quad (38)$$

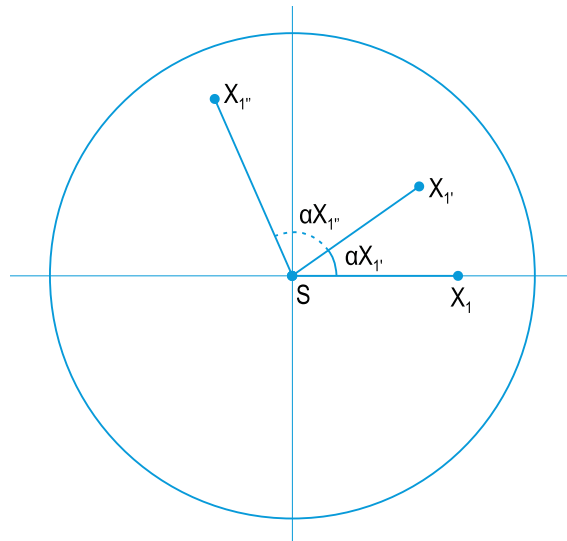


Figura 28 - Ordem do Ângulo α

Fonte: Elaborada pelo Autor

Desta forma, a generalização de P_{random} para o estabelecimento de comunicação entre o nó sensor x_2 , posicionado fora do alcance do nó sensor central S, e o nó sensor central S, através do nó sensor x_1 , posicionado dentro do alcance do nó sensor central S, é calculada conforme a equação (39).

$$P_{random} = \frac{\sum \text{Areas acrescentadas} + \text{Area fundamental}}{A_{SR}} \quad (39)$$

Em que A_{SR} é a área de sensoriamento e é calculada por: $A_{SR} = \pi R^2$

4.5.4 Metodologia para Cálculo do Parâmetro K2

Conforme definido e apresentado na seção 4.5.1 Caso 1 – Alta Velocidade de Deslocamento do Nó Sensor a probabilidade P_{random} é o próprio parâmetro K2 para o caso em que os nós sensores se movem em alta velocidade, isto é, a distância percorrida (D_p) em Δt ($t_1 - t_0$), for maior que a distância de alcance de transmissão deste mesmo nó sensor.

Para o caso 2, definido na seção 4.5.2 Caso 2 – Baixa Velocidade de Deslocamento do Nó Sensor em que os nós sensores se movem em baixa velocidade, a probabilidade P_{random} é um dado de entrada na equação (20).

Como definido na seção 4.5.3.1 Generalização de Prandom, é necessária a realização do cálculo das áreas acrescentadas pelos nós sensores, as quais dependem dos parâmetros r e ρ . O parâmetro r é uma variável definida e conhecida e o parâmetro ρ é estabelecido com base nas coordenadas (x,y) de cada nó sensor, de forma que $\rho = \sqrt{x^2 + y^2}$, sendo que estas coordenadas são sorteadas de forma aleatória para cada nó sensor.

Uma vez que os valores de K_2 tenham sido obtidos, os mesmos serão utilizados juntamente com o parâmetro K_1 , como dados de entrada para o algoritmo de transição de estados, para o caso de deslocamento em baixa velocidade, permitindo assim, que o grau de sincronismo seja estabelecido em função da taxa de envio de mensagens, da relação entre nós ativos e nós totais e da taxa de mobilidade dos nós dentro da área de sensoriamento.

4.6 Algoritmo de Transição de Estados

Quando um determinado nó sensor se desloca em baixa velocidade é necessário determinar a transição de estados deste nó sensor em um determinado instante de tempo. Para tanto, é proposto um algoritmo com o objetivo de determinar qual estado o nó sensor estará no instante de tempo t_1 dado o seu estado no instante de tempo t_0 .

Este algoritmo está baseado na comparação entre uma variável aleatória (Z) e a probabilidade de mobilidade do nó (K_2), e na comparação entre uma segunda variável aleatória (Y) e a relação entre quantidade de nós sensores ativos e quantidade de nós sensores totais (K_1), que representa o consumo de energia.

O algoritmo deve ser iniciado considerando que o nó sensor está no estado *HABILITADO* e então são sorteados dois valores aleatórios para que as comparações sejam realizadas.

A Tabela 1 apresenta o estado atual de um nó sensor, as condições para determinar o próximo estado (valores de X , Y e comparações com os parâmetros K_1 e K_2) e o novo estado que o nó sensor deve assumir no instante de tempo seguinte.

Tabela 1 - Transição dos Estados

ESTADO ATUAL	Z	Y	NOVO ESTADO
HABILITADO	$Z > K_2$	-	FORA DO ALCANCE
	$Z \leq K_2$	$Y < K_1$	NÃO HABILITADO
		$Y \geq K_1$	PERMANECE HABILITADO
NÃO HABILITADO	-	$Y > K_1$	HABILITADO
		$Y \leq K_1$	PERMANECE NÃO HABILITADO
FORA DO ALCANCE	$Z \geq K_2$	-	PERMANECE FORA DO ALCANCE
	$Z < K_2$		HABILITADO

Fonte: Elaborada pelo Autor

O algoritmo que representa a situação de transição dos estados é descrito a seguir

Seja:

$K_1 = f$ (Grau de Sincronismo), em que $0 \leq K_1 \leq 1$

e

$K_2 = f$ (Mobilidade da Rede), em que $0 \leq K_2 \leq 1$

FAÇA ENQUANTO FOR VERDADE:

{

Quando o nó estiver no estado HABILITADO

$Z = \text{random}(0,1)$

Se $X > K_2 \Rightarrow$ FORA DO ALCANCE

Senão

$Y = \text{random}(0,1)$

Se $Y < K_1 \Rightarrow$ NÃO HABILITADO

$Y \geq K_1 \Rightarrow$ PERMANECE HABILITADO

Quando no estado NÃO HABILITADO

$Y = \text{random}(0,1)$

Se $Y > K_1 \Rightarrow$ HABILITADO

Se $Y \leq K_1 \Rightarrow$ PERMANECE NÃO HABILITADO.

Quando no estado FORA DO ALCANCE

$Z = \text{random}(0,1)$

Se $Z \geq K_2 \Rightarrow$ PERMANECE FORA DO ALCANCE

$Z < K_2 \Rightarrow$ HABILITADO

5 IMPLEMENTAÇÃO DO SIMULADOR

Este capítulo apresenta a implementação do modelo desenvolvido no [capítulo 4](#), para os casos de mobilidade em alta velocidade. Para tanto foi desenvolvido um script para cálculo das perdas de comunicação no sistema e estes dados, juntamente com as frequências de oscilação dos relógios *master* e *slave* e de operação do sistema, são base de entrada do simulador implementado na ferramenta *Simulink*®.

5.1 Descrição do Simulador

O modelo de simulação construído, apresentado na Figura 29, é composto de quatro componentes principais: *Master*, *Message Loss*, *Network Delay* e *Slave*. Cada componente tem uma função específica na operação do modelo e pode ser composto por um ou mais blocos.

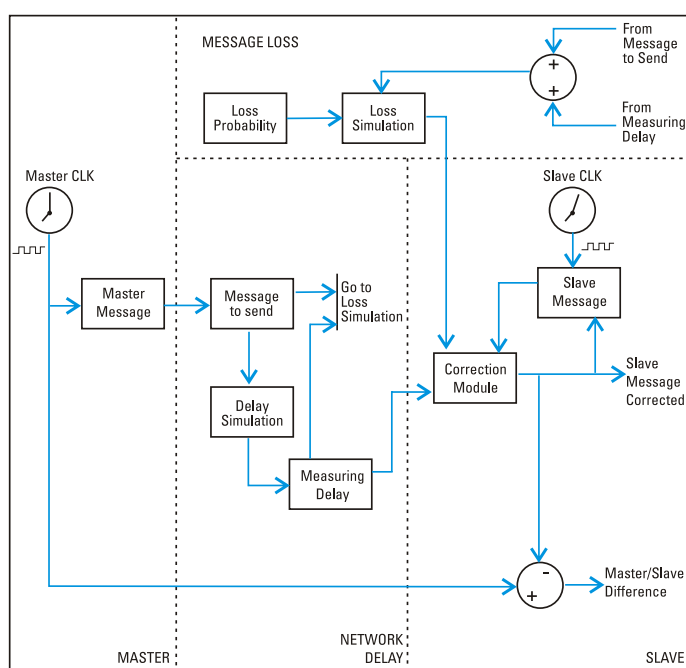


Figura 29 - Diagrama de Blocos em Alto Nível

Fonte: Elaborada pelo Autor

5.1.1 Componente Master

O componente *Master* é composto pelos blocos *Master_CLK* e *Master_Message*. A principal função deste componente é a geração do relógio de referência para o sistema a partir de um gerador de pulso e simular o relógio de referência (*Primary Reference Clock - PRC*) a partir do qual todo o sistema deverá ser sincronizado.

5.1.2 Componente Message Loss

O componente *Message Loss* é composto de dois blocos: *Loss_probability* e *Loss_simulation*. A principal função deste componente é simular a perda de mensagens enviadas pelo bloco *Message_to_send*. Desta forma o componente *Slave*, através do bloco *correction_module* calcula o ajuste que deve ser realizado no *Slave_CLK*. A Figura 30 apresenta o detalhamento do componente *Message_Loss*.

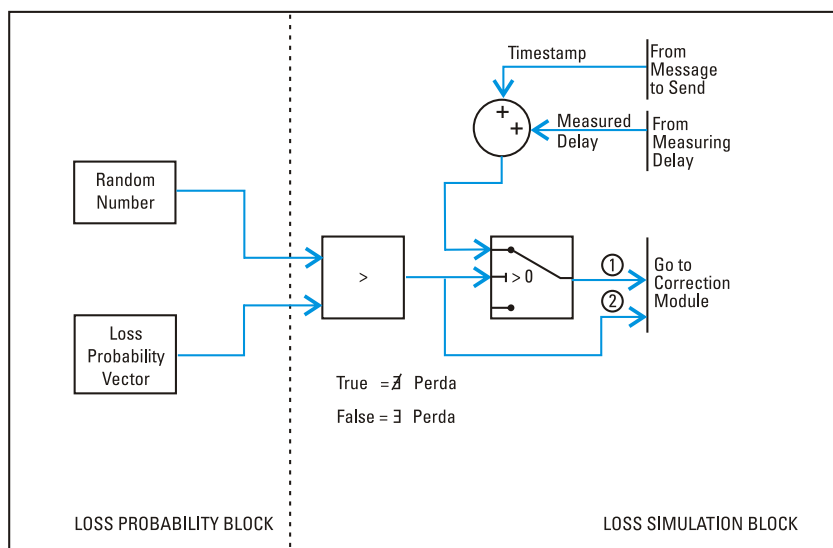


Figura 30 - Componente Message Loss

Fonte: Elaborada pelo Autor

5.1.3 Componente Network Delay

O componente *Network Delay* é composto pelos blocos *Message_to_Send*, *Delay_Simulation* e *Measuring_Delay*. Este componente não foi construído para emular o comportamento de equipamentos de rede ou canais de comunicação. A função principal deste componente é calcular o atraso a que os pacotes estão submetidos durante a transmissão e atualizar o componente *Slave* com esta informação. Isto é realizado através dos blocos *Delay_Simulation* e *Measuring_Delay* que inserem atrasos aleatórios simulando o comportamento de uma rede de pacotes. Os blocos *Message_to_Send*, *Delay_Simulation* and *Measuring_Delay* são apresentados em detalhes na Figura 31.

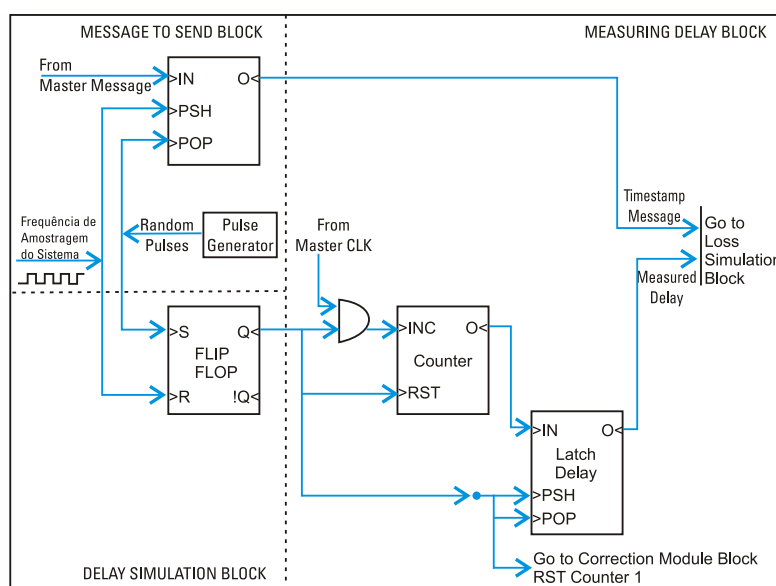


Figura 31 - Componente Network Delay

Fonte: Elaborada pelo Autor

Os blocos *Message_to_Send* and *Delay_Simulation* simulam o atraso em uma rede de pacotes. Isto é realizado através do componente gerador de pulso com intervalos aleatórios. Estes intervalos aleatórios são obtidos através do uso de diferentes distribuições probabilísticas.

O objetivo desta abordagem é que escala de tempo do modelo passa a ser definida pelo gerador de pulsos de referência e não pelo passo da simulação. Conseqüentemente, o valor aleatório gerado tem de ser convertido em um intervalo

de tempo no *range* do relógio de referência e não pode ser usado como número dos passos de simulação. A Figura 32 mostra o modelo utilizado para a conversão de escalas de tempo. Ao final do intervalo aleatório, um pulso é gerado para carregar o pacote no componente *Slave*. Através do componente Gerador de Pulsos também é possível alterar a distribuição probabilística e com isto analisar diferentes tipos de aplicações em rede.

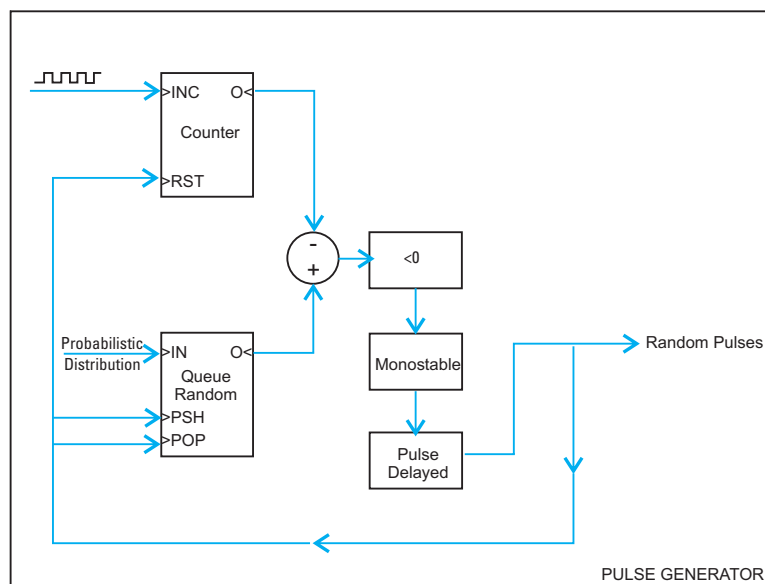


Figura 32 - Geração de Pulsos Aleatórios na Escala de Tempo do Modelo

Fonte: Elaborada pelo Autor

5.1.4 Componente Slave

O componente *Slave* é composto pelos blocos *Slave_CLK*, *Slave_Message* e *Correction_Module*. A função principal deste componente é manter o bloco *Slave_Message* ajustado com o relógio gerado pelo componente *Master*.

O bloco *Correction_Module*, apresentado na Figura 33 em detalhes, tem como função simular a correção a ser aplicada no *Slave_CLK* considerando o atraso da mensagem gerada pelo componente *Master* e a perda de mensagem simulada pelo componente *Message Loss*.

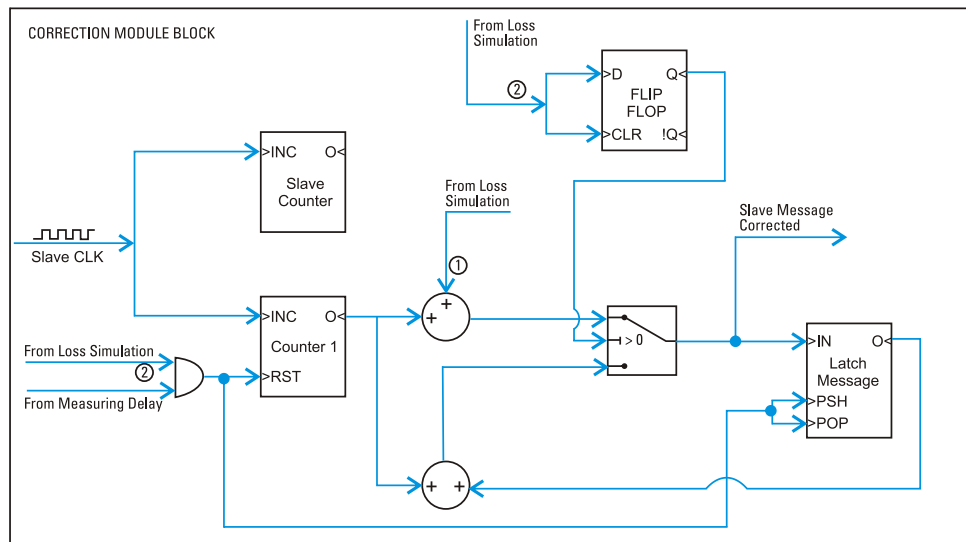


Figura 33 - Bloco Correction Module

Fonte: Elaborada pelo Autor

5.2 Modo de Operação do Modelo

Para a inicialização do sistema o mesmo deve ser parametrizado com alguns dados. São eles: frequência de oscilação nos blocos *Master_CLK* e *Slave_CLK*, frequência de amostragem do sistema e o vetor de probabilidade de perda da mensagem, o qual é inserido no componente *Message Loss*. Este vetor é resultado de um script de simulação de probabilidade executado previamente.

5.2.1 Script de Simulação

Para analisar a probabilidade de um determinado nó sensor perder comunicação com o nó sensor sink foi desenvolvido um software em MATLAB baseado no método de Monte Carlo. Este método de simulação estatística utiliza sequências de números aleatórios para desenvolver simulações. Em outras palavras, é visto como método numérico universal para resolver problemas por meio de amostragem aleatória (aproximação da solução).

O método torna desnecessário escrever as equações diferenciais que descrevem o comportamento de sistemas complexos. A única exigência é que o sistema físico ou matemático seja descrito (modelado) em termos de funções de densidade de distribuição de probabilidade (FDP). Uma vez conhecidas essas distribuições, a Simulação de Monte Carlo pode proceder fazendo as amostragens aleatórias a partir das mesmas. Este processo é repetido inúmeras vezes e o resultado desejado é obtido por meio de técnicas estatísticas sobre um determinado número de realizações (amostra) que podem variar de dezenas a milhões de vezes (NASSER, 2012).

Como resultado deste script pode se obter uma probabilidade de perda ou um vetor de probabilidade de perda para cada vez que o script é executado com os mesmos parâmetros de entrada. O vetor de probabilidade de perda é constituído da probabilidade de perda obtida para cada execução, isto é, se a quantidade de execuções for maior que 1, o vetor é formado. O código de implementação do script está descrito no APÊNDICE B – SCRIPTS MATLAB.

Para que a simulação seja realizada são necessários cinco parâmetros de entrada, descritos a seguir:

- 1) raio da área de sensoriamento, definido no software como variável *raio_area*;
- 2) raio de alcance de transmissão dos nós sensores, variável *raio_alcance*;
- 3) quantidade de nós sensores a serem distribuídos aleatoriamente na área de sensoriamento, variável *qtd_sensores*;
- 4) parâmetro percentual de nós sensores ativos, variável *A* (ex: se *A* for igual a 1, então todos os nós sensores estarão ativos) e
- 5) número de execuções da simulação para utilização do método de Monte Carlo, variável *qtd_execucoes*.

O software está dividido em três grandes blocos de execução. O primeiro bloco define o posicionamento dos nós sensores com base em um sorteio aleatório. Para tanto é sorteado um par de coordenadas (x,y). Para cada par de coordenada sorteada é realizado uma verificação para determinar se o nó sensor está dentro ou fora da área de sensoriamento (Asr) de raio R . Se o nó estiver fora da área de sensoriamento ele é descartado e um novo sorteio é realizado. Se o nó estiver dentro da área da área de sensoriamento então as coordenadas são armazenadas em uma matriz, denominada Matriz de Nós Ativos.

O segundo bloco tem como função, verificar quais destes nós sensores distribuídos aleatoriamente se comunicam diretamente com o nó sensor sink e quais nós sensores não se comunicam (teste nós sensores ativos x nó sensor sink – Figura 35).

Após este primeiro teste os nós sensores são separados em dois grupos, denominados C0 (nós sensores que se comunicam diretamente com o nó sensor sink) e NC1 (nós sensores que não se comunicam diretamente com o nó sensor sink). O terceiro e último bloco de execução verifica se os nós sensores do grupo NC1 se comunicam com os nós sensores do grupo C0.

Para que seja considerada a existência de comunicação é necessário que um determinado nó sensor pertencente ao grupo NC1 esteja dentro do raio de alcance de transmissão de um nó sensor pertencente ao grupo C0, conforme Figura 34.

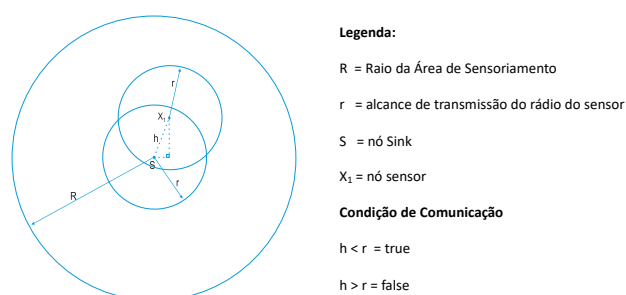


Figura 34 - Validação de Comunicação entre dois nós sensores

Fonte: Elaborada pelo Autor

Este teste de comunicação ocorre nó a nó. Isto é, suponha que os nós sensores 1 a 5 se comuniquem diretamente com o nó sensor sink (pertencem ao grupo C0) e os nós 6 a 10 não se comuniquem diretamente com o nó sensor sink (pertencem ao grupo NC1). Então o software utilizará as coordenadas do nó 6 para validar se existe comunicação deste nó sensor 6 com algum nó sensor pertencente ao grupo C0. O primeiro teste a ser realizado será nó sensor 6 contra nó sensor 1, o segundo teste será nó sensor 6 contra nó sensor 2 e assim por diante.

Uma vez que esta condição de teste seja verdadeira, isto é, exista comunicação, o loop de execução é interrompido, as coordenadas do nó sensor são armazenadas em C1 (grupo de nós que possui comunicação com sink através de algum nó pertencente ao grupo de nós sensores C0) e inicia-se a validação do próximo nó sensor pertencente ao grupo NC1 (ex: nó sensor 7 contra nó sensor 1, nó 7 sensor contra nó sensor 2 e assim por diante). Se um determinado nó sensor pertencente ao grupo

NC1 não tiver a condição de comunicação verdadeira com pelo menos 1 nó sensor pertencente ao grupo C0, isto significa que este nó sensor está posicionado fora do raio de alcance dos nós sensores pertencentes ao grupo C0 e suas coordenadas serão armazenadas em NC2 (nós sensores que não possuem comunicação com o nó sensor sink, seja diretamente ou através de outro nó sensor pertencente ao grupo C0), o qual será testado contra os nós sensores pertencentes ao grupo C1, conforme representado na Figura 35.

Este loop de teste de comunicação é executado até que se teste a condição de comunicação de todos os nós sensores distribuídos na Área de Sensoriamento, resultando em uma condição final de quantos nós sensores se comunicam o nó sensor sink, independente de quantos saltos sejam necessários, e quantos nós sensores não se comunicam com o nó sensor sink.

Como resultado de saída desta execução tem-se a perda para cada execução ($qtd_execucoes = 1$) ou o vetor de perda ($qtd_execucoes > 1$) e para os nós sensores que possuem comunicação com o nó sensor sink, o software calcula quantos saltos foram necessários para o estabelecimento desta comunicação.

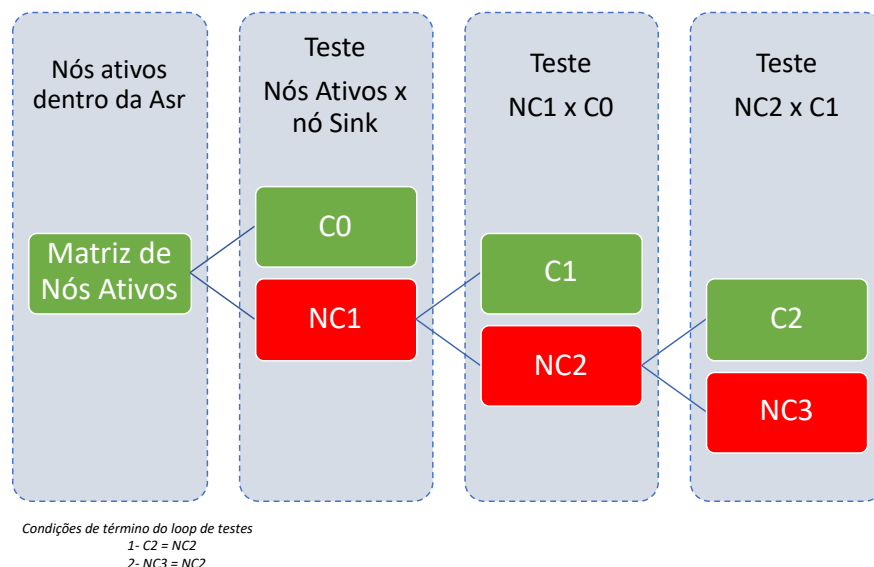


Figura 35 - Lógica do Teste de Comunicação

Fonte: Elaborada pelo Autor

5.2.1.1 Validação do Script de Simulação

Para a validação do script de simulação foi realizada a comparação dos resultados obtidos através da execução do script com os resultados calculados matematicamente conforme capítulo 4.

Considerando dois nós sensores, X_1 e X_2 com alcance de transmissão de raio r , distribuídos aleatoriamente em uma área de sensoriamento circular A_{SR} , de raio R e um nó sensor sink S , posicionado no centro da área A_{SR} , existe uma Probabilidade $P(E_i)$ de ocorrer 4 possíveis eventos, conforme descritos Tabela 2.

Tabela 2 - Eventos Possíveis

Evento	Sensor X_1	Sensor X_2	Descrição
E_1	$X_1 \nrightarrow S$	$X_2 \nrightarrow S$	X_1 e X_2 não se comunicam com S
E_2	$X_1 \rightarrow S$	$X_2 \nrightarrow S$ $X_2 \nrightarrow X_1$	X_1 se comunica com S X_2 não se comunica com X_1 nem com S
E_3	$X_1 \rightarrow S$	$X_2 \rightarrow X_1$	X_1 se comunica com S e X_2 se comunica com X_1
E_4	$X_1 \rightarrow S$	$X_2 \rightarrow S$	X_1 e X_2 se comunicam com S

Fonte: Elaborada pelo Autor

Para cada evento constante da Tabela 2 foram realizadas simulações utilizando o software desenvolvido e descrito [seção 5.2.1 Script de Simulação](#). Os resultados obtidos foram comparados com os resultados obtidos a partir da resolução matemática desenvolvida e apresentada na equação (28).

A Tabela 3 apresenta os parâmetros de entrada utilizados para a realização da execução do script para validação.

Tabela 3 - Parâmetros de Validação

Parâmetro	Valor
<i>raio_area</i>	$R = 4r$
<i>raio_alcance</i>	r
<i>qtd_sensores</i>	2 sensores X_1 e X_2 , mais nó sink
A_{SR}	1 (todos os nós estão ativos)

Fonte: Elaborada pelo Autor

O evento E_3 é modelado matematicamente no capítulo 4 e sua extrapolação foi implementada em software, conforme descrito na seção 5.2.1 Script de Simulação, de forma que um determinado nó sensor X_i possa estabelecer comunicação com o nó sensor sink através de n saltos. Desta forma para a validação do script, foram realizadas 22 simulações, onde para cada simulação o número de repetições foi variado. Sendo que a primeira simulação teve 8000 repetições e a última simulação foi executada com 60000 repetições (método de Monte Carlo).

A probabilidade obtida para cada simulação realizada é apresentada na Figura 36 e comparada com a probabilidade calculada. Os resultados calculados e simulados para os demais eventos, bem como a média dos resultados simulados obtidos para o evento 3 (E_3) são apresentados na Tabela 4.

Tabela 4 - Probabilidade por Evento – Calculado x Simulado

Evento	Resultado Calculado	Resultado Simulado
E ₁	87,890 %	87,733 %
E ₂	11,331 %	11,502 %
E ₃	0,389 %	0,390 %
E ₄	0,390 %	0,375 %

Fonte: Elaborada pelo Autor

O evento analisado na Figura 36 é o E_3 , uma vez que este evento representa a probabilidade de um nó sensor se comunicar com o nó sensor sink através de outro nó sensor ($X_2 \rightarrow X_1 \rightarrow S$). Conforme observado no gráfico apresentado e na Tabela 4, a probabilidade média obtida com a simulação realizada no software desenvolvido foi de 0,390% enquanto que a probabilidade calculada matematicamente, através da equação (28) é de 0,389%. Isto demonstra que é possível utilizar o software para obter resultados de simulações para cenários extrapolados, isto é, com quantidades superiores a 2 nós sensores, sem a necessidade de desenvolvimento de modelos matemáticos complexos.

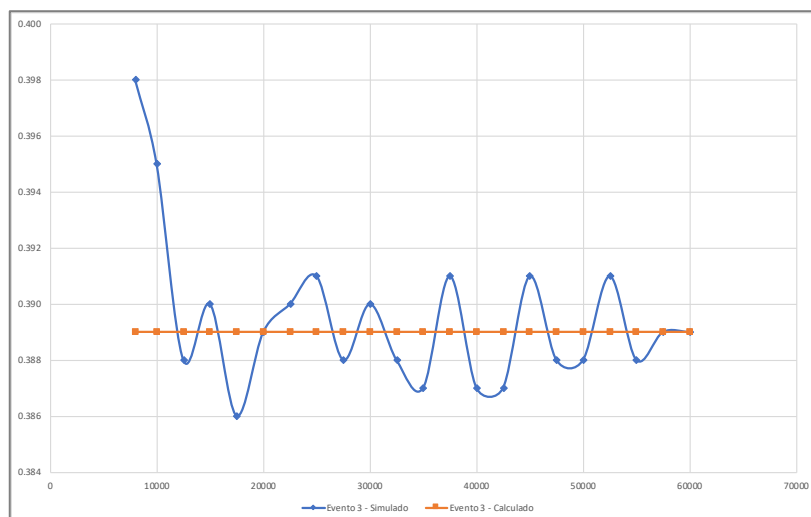


Figura 36 - Probabilidade de ocorrência do Evento E_3 ($X_2 \rightarrow X_1 \rightarrow S$)

Fonte: Elaborada pelo Autor

5.2.2 Funcionamento do Simulador

Para iniciar a operação do simulador é necessário parametrizar as frequências internas dos osciladores do *Master_CLK* e do *Slave_CLK*, bem como é necessário inserir o vetor de perda, *loss_probability_vector*, no componente *Message Loss*, bloco *Loss_probability*. Este vetor é obtido a partir da execução do script de simulação descrito na [seção 5.2.1 Script de Simulação](#).

Os blocos *Master_CLK* e *Slave_CLK*, no instante de tempo t_0 , iniciam seus contadores internos baseados em sinais independentes (cada bloco possui um oscilador interno). No mesmo instante de tempo t_0 o gerador de pulso aleatório é disparado e o vetor de probabilidade perda é carregado no sistema no bloco *Loss_probability*, onde, no bloco *Loss_Simulation*, cada valor deste vetor é comparado com um gerador de números aleatórios. Esta comparação resulta em duas condições:

- comparação é Falsa: então o número aleatório gerado é menor que o valor da perda e portanto existe perda de mensagem de sincronismo gerada pelo bloco *Master_CLK*;

- comparação é Verdadeira: então o número aleatório gerado é maior que o valor da perda e portanto não existe perda de mensagem de sincronismo gerada pelo

bloco *Master_CLK*. Esta mensagem poderá ou não sofrer atraso de transmissão, esta condição depende do gerador de pulsos aleatórios do bloco *Delay_Simulation*.

O oscilador interno do *Slave_CLK* é ajustado em uma frequência diferente da frequência interna do oscilador interno do *Master_CLK*, com o objetivo de se obter uma diferença entre o *Master_CLK* e o *Slave_CLK*, simulando assim a existência de um relógio de referência - PRC, neste caso o *Master_CLK*. A informação gerada no bloco *Master_CLK* é periodicamente lida e inserida na fila do bloco *Message_to_Send* a uma taxa constante, a qual representa a taxa de amostragem do sistema.

Quando não há perda de mensagem, ela é inserida na fila do bloco *Message_to_Send* é armazenada no seu *buffer* até o instante de tempo t em que um pulso aleatório é gerado. O gerador de pulsos de controle é apresentado na Figura 37.

O pulso aleatório gera o sinal *remove* da fila do bloco *Message_to_Send* fazendo com que a mensagem (*Timestamp_Message*), armazenada na fila seja enviada para o bloco *Correction_Module*. A duração deste pulso é ajustada para permitir que um novo processo de medição seja iniciado mantendo assim o atraso com a distribuição probabilística desejada.

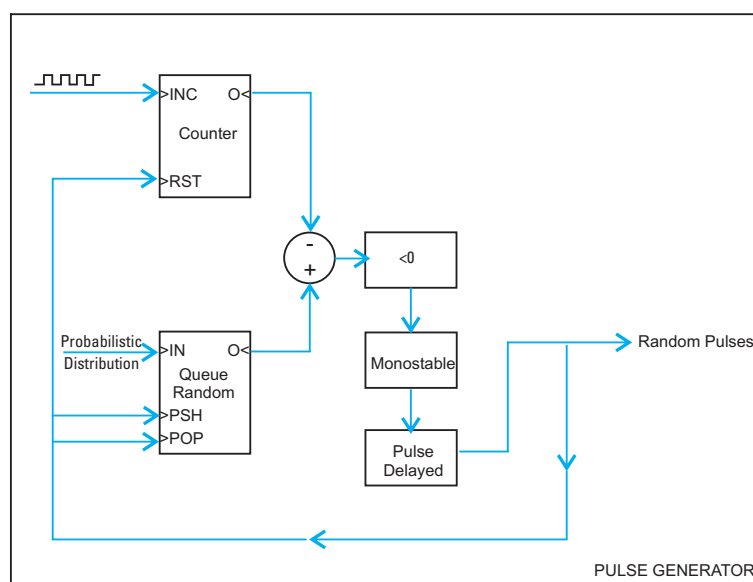


Figura 37 - Gerador de Pulso de Controle

Fonte: Elaborada pelo Autor

O pulso aleatório gerado também dispara um contador que tem como função medir o atraso, quando ocorrer, gerado na escala de tempo do modelo. Este atraso é dado

pela diferença de valores entre a entrada e a saída da fila de mensagens do bloco *Message_to_Send* e é medido com base nos pulsos do relógio de referência. Quando a mensagem armazenada no bloco *Message_to_Send* é transmitida, o valor do atraso é armazenado no bloco *Latch_Delay Measuring_Delay*. Caso não exista atraso, o valor armazenado é zero.

Como resultado do algoritmo implementado, a mensagem *Timestamp_Message* gerada pelo bloco *Message_to_Send* e a mensagem *Measured_Delay* gerada pelo bloco *Measuring_Delay* são enviadas para o bloco *Correction_Module*.

O bloco *Correction_Module* realiza a soma dos valores recebidos através das mensagens *Timestamp_Message* e *Measured_Delay*, as quais o bloco *Loss_simulation* permite o envio uma vez que não houve perda, e depois adiciona este resultado ao valor do contador *Counter_1*. Este contador inicia sua operação em t_0 com uma frequência ligeiramente diferente da frequência do *Master_CLK*, porém este contador é reiniciado sempre que o *Latch_Delay* do bloco *Measuring_Delay* fornece a mensagem *Measured_Delay*. Com isso é possível determinar o valor correto do *Slave_CLK* e, assim, corrigi-lo com os valores recebidos, obtendo-se assim a mensagem *Slave_Corrected_Message*. Esta mensagem é então comparada com o valor do *Master_CLK* e, então, o valor da correção estabelecida pelo sistema é obtida.

Quando existe perda de mensagem a diferença de operação ocorre no bloco *Correction_module*, uma vez que é necessário que o sistema saiba qual foi a mensagem recebida pelo componente *Slave* em $t-1$, a qual é armazenada no *Latch_message* e mantenha-se atualizado tendo como referência este valor somado ao valor gerado pelo seu oscilador interno, o qual reiniciou sua contagem no instante de tempo $t-1$ (última mensagem recebida), até o instante de tempo em que uma nova mensagem chegar.

A partir deste momento, o *Slave* incrementa o seu relógio com base no seu oscilador local até o instante de tempo em que uma nova mensagem contendo a referência de sincronismo do *Master_CLK* seja recebida e o algoritmo de correção seja novamente executado.

O bloco *Correction_Module* é ainda composto pelo *Slave_Counter* cuja função é a de apenas permitir a comparação com *Master_CLK*, permitindo assim que a leitura dos valores *Master_CLK* e *Slave_CLK* sem correção seja realizada.

5.3 Validação do Simulador

Para a validação do simulador foram estabelecidas duas condições de contorno relacionadas a perda de mensagem. Com os resultados obtidos foram analisados os parâmetros MTIE e atraso entre o *Master_CLK* e o *Slave_CLK*. Para as condições de validação não é necessário executar o script de simulação, pois o valor do vetor de perda é a variável a ser controlada para a validação do simulador e, portanto, é fixado diretamente no simulador.

5.3.1 Validação com Perda

Para este cenário o vetor de perda, *loss_probability_vector* no componente *Message Loss*, bloco *Loss_probability* foi parametrizado com o valor 1, isto é, todas as mensagens geradas pelo *Master_CLK* são perdidas e portanto o valor do *Slave_CLK* corrigido deve ser o próprio valor gerado pelo oscilador interno do *Slave_CLK*.

As frequências dos osciladores internos do *Master_CLK* e do *Slave_CLK* foram ajustadas de forma que seja gerada uma diferença entre os relógios *master* e *slave* de 0,002 a cada passo de integração do simulador. O tempo de execução do simulador foi parametrizado em 5 segundos com passo de integração de 0,000001, isto resulta, ao final da simulação, em uma diferença entre os relógios no valor 10000, conforme a equação (40).

$$Atraso\ total = (\Delta freq) \times \frac{\text{tempo de simulação}}{\text{passo de integração}} \quad (40)$$

Em que $\Delta freq$ é a diferença entre as frequências *master* e *slave*.

A Figura 38 e a Figura 39 apresentam respectivamente o cálculo do MTIE e o cálculo do atraso para a situação de perda total de mensagens. Para o cálculo do parâmetro MTIE foi considerado $\tau_0 = 5$ e $n = \{1, 5000\}$.

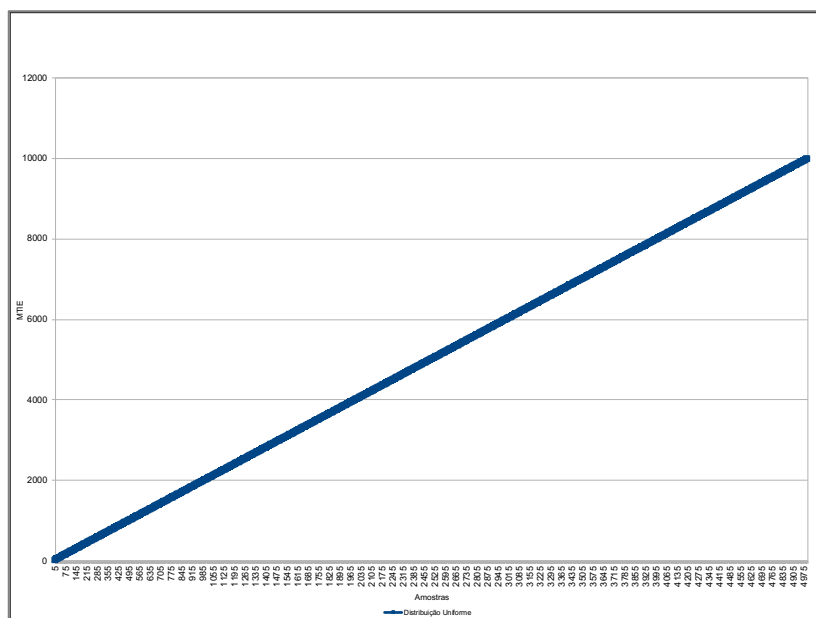


Figura 38 - Cálculo MTIE - Perda 100%

Fonte: Elaborada pelo Autor

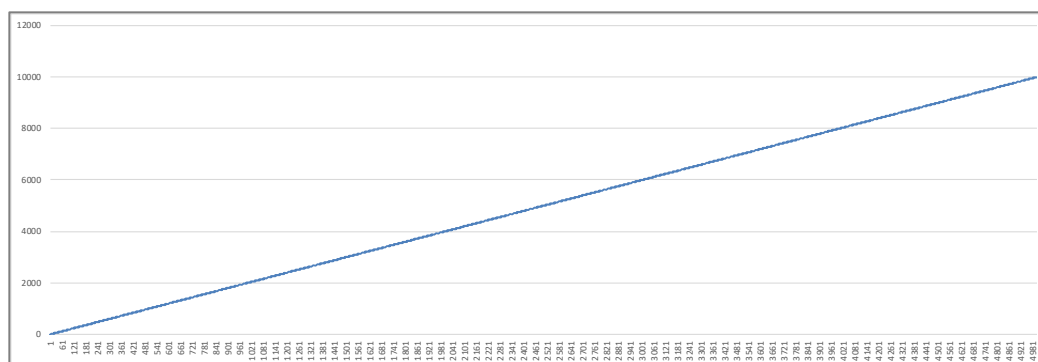


Figura 39 - Atraso entre Master e Slave - Perda 100%

Fonte: Elaborada pelo Autor

O valor do atraso cresce a cada passo de integração da simulação executada a uma taxa constante. Esse comportamento está de acordo com o esperado uma vez que todas as mensagens geradas pelo *Master_CLK* são perdidas e, portanto, o *Slave_CLK* será sempre atualizado com o valor de mensagem gerada internamente pelo seu próprio oscilador. Nesta situação o MTIE e o atraso também devem apresentar o mesmo comportamento, uma vez que o MTIE é o atraso máximo obtido em um determinado intervalo de tempo para um determinado conjunto de amostras. Como o atraso é sempre crescente o valor do MTIE também cresce a cada passo de integração com uma taxa constante. Este comportamento é justamente o obtido na

execução do simulador e pode ser observado quando comparada a Figura 38 com a Figura 39. Ao final da simulação o valor do atraso obtido foi de 10000, conforme valor calculado a partir da equação (40).

5.3.2 Validação sem Perda

Para este cenário o vetor de perda, *loss_probability_vector* no componente *Message Loss*, bloco *Loss_probability* foi parametrizado com o valor 0, isto é, todas as mensagens geradas pelo *Master_CLK* são enviadas e estão sujeitas a atraso. Portanto o valor do *Slave_CLK* corrigido considera o valor do atraso, o valor da mensagem recebida e o valor gerado pelo oscilador interno do *Slave_CLK*.

Da mesma forma que o cenário anterior as frequências dos osciladores internos do *Master_CLK* e do *Slave_CLK* foram ajustadas de forma que seja gerada uma diferença entre os relógios *master* e *slave* de 0,002 a cada passo de integração do simulador. O tempo de execução do simulador foi parametrizado em 5 segundos com passo de integração de 0,000001. Porém como não existe perda de mensagem é esperado que o valor do atraso final seja inferior ao atraso obtido no caso anterior. Também é esperado que o MTIE tenha um comportamento degrau, isto é, existam intervalos de tempo onde o atraso se mantém constante, conforme previsto nas normas do ITU-T (ITU-T G.8261, 2008).

Para obter as mensagens de atraso na rede e realizar as simulações foram utilizados dois tipos de distribuição probabilística: distribuição uniforme e distribuição exponencial. Desta forma foi possível validar o comportamento do simulador durante a transmissão de dados em uma rede de pacotes.

A simulação de uma distribuição de probabilidade uniforme tem como objetivo analisar o comportamento do sistema modelado em um ambiente em que a rede de pacotes possui tráfego controlado como, por exemplo, um sistema de rede de sensores distribuídos em uma fábrica, onde é possível determinar e/ou controlar o momento em que cada transmissão ocorre, bem como o tamanho de cada mensagem a ser transmitida.

Os parâmetros de distribuição do número aleatório gerado dependem do atraso médio que se pretende simular e da frequência desejada na geração das mensagens. O período de medição é determinado pelo produto do atraso médio (N) pelo período de geração dos números aleatórios (T), que é:

$$N.T = \text{Período de Medição (41)}$$

A simulação realizada com a distribuição de probabilidade uniforme utiliza um período de medição de 0,01 e média de atraso de 50 μ s para uma unidade de tempo de simulação igual a 1 μ s, resultando em $T = 0,0002$. Isto garante que nenhum valor amostrado pelo sistema seja perdido, isto é, não seja armazenado na fila do bloco *Message_to_Send*.

A simulação de uma distribuição de probabilidade exponencial tem como objetivo analisar o comportamento do sistema modelado em um ambiente não controlado, onde o atraso que cada pacote sofre não tem nenhuma dependência em relação ao atraso anterior e, conseqüentemente, não podem ser controlados pelo emissor ou receptor da mensagem. Os parâmetros utilizados na simulação são: atraso médio $N = 0,1$ e período de medição de 0,01, resultando em uma frequência de operação de 10 KHz. Assim, a geração média de números aleatórios é menor do que a taxa de operação do sistema, garantindo assim que todos os valores da amostra sejam inseridos na fila do bloco *Message_to_Send*.

De forma análoga à validação com perda, foi calculado o parâmetro MTIE e o atraso entre o *Master_CLK* e o *Slave_CLK* para a situação sem perda de mensagens. Ambas as situações são apresentadas na Figura 40 e Figura 41, respectivamente. Para o cálculo do parâmetro MTIE foram considerados considerado $\tau_0 = 5$ e $n = \{1,100\}$.

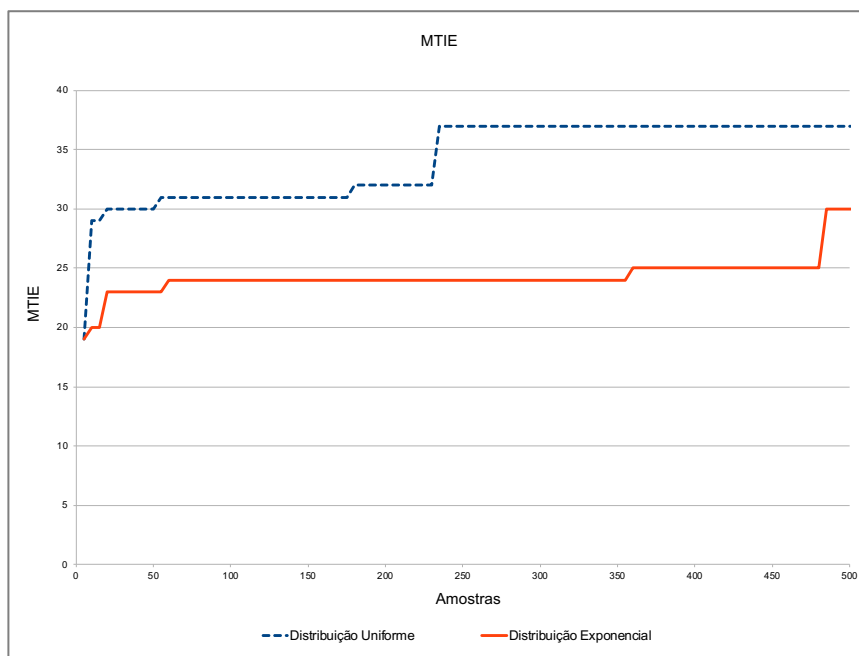


Figura 40 - Cálculo MTIE - Sem Perda

Fonte: Elaborada pelo Autor

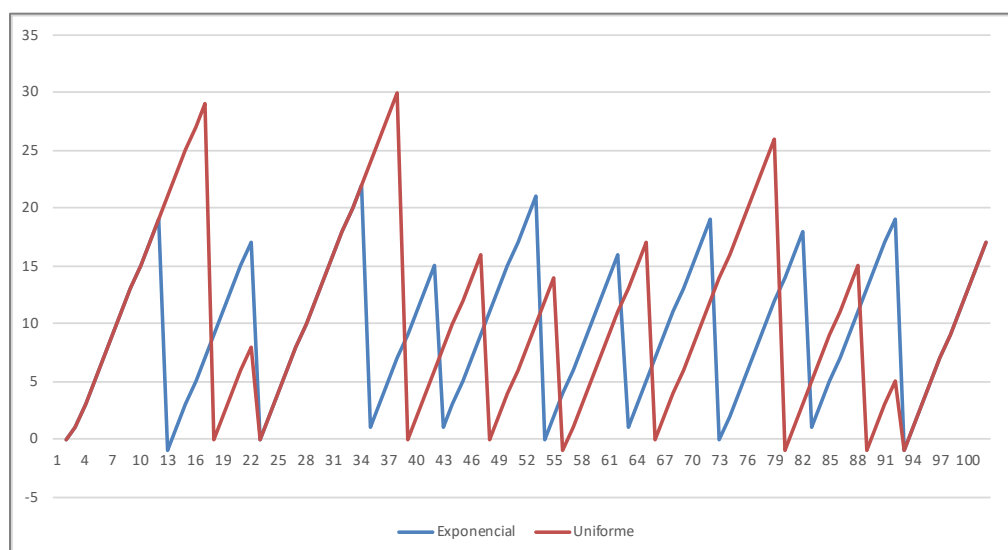


Figura 41 - Atraso entre Master e Slave – Sem Perda

Fonte: Elaborada pelo Autor

Os gráficos apresentados acima, demonstram o correto funcionamento do simulador implementado onde a correção do tempo no *Slave_CLK* é efetuada de acordo com a recepção das mensagens enviadas pelo *Master_CLK*. No intervalo de tempo entre duas mensagens do *Master_CLK* o *Slave_CLK* mantém sua marcação de tempo baseado no seu próprio oscilador interno. Como este oscilador é de menor precisão

que o oscilador do *Master_CLK* (PRC) o mesmo apresenta uma diferença de tempo em relação ao PRC até o momento em que recebe uma nova mensagem de sincronismo. A variação de atraso se dá devido ao atraso no envio das mensagens ser simulado através de distribuições probabilísticas.

O comportamento do parâmetro MTIE, observado no gráfico da Figura 40 é o comportamento padrão esperado para este tipo de medida, conforme especificado na norma ITU-T G.8261 (ITU-T G.8261, 2008). O cálculo de MTIE apresenta ainda o atraso máximo para o intervalo de amostras definido para as distribuições de probabilidade uniforme e exponencial. Em ambas as situações o *Slave_CLK* tem sua correção efetuada evidenciando o correto funcionamento do simulador implementado, validando assim, a proposição de que um sistema de sincronização baseado na transmissão de marcas de tempo (*timestamps*) em pacotes e o ajuste do relógio de menor precisão (*Slave_CLK*) em função destas marcas de tempo pode ser utilizado para a realização de sincronização de redes de sensores.

6 SIMULAÇÕES E RESULTADOS

As simulações realizadas tem por objetivo analisar o sincronismo em uma rede de sensores móveis tendo como base o modelo matemático desenvolvido e sua implementação em *Simulink*®. Para a análise deste sincronismo foram consideradas as seguintes premissas:

- sincronismo baseado em troca de mensagens;
- área de sensoriamento circular de raio R ;
- alcance de transmissão do sensor com raio r . Uma vez que um determinado nó sensor X_2 esteja dentro do raio de alcance de outro nó sensor X_1 , o nó sensor X_2 esta apto a receber mensagens;
- distribuição aleatória dos nós sensores dentro da área de sensoriamento;
- quantidade de sensores ativos;
- mobilidade em alta velocidade dos nós sensores.

Para todas as simulações executadas é considerado que a taxa de perda de comunicação é igual a probabilidade de não comunicação resultante da execução do script de simulação. Conforme demonstrado na [seção 4.5 Modelagem da Mobilidade](#) esta consideração é possível uma vez que as simulações executadas consideram que todos os nós sensores se movimentam em alta velocidade.

A quantidade de saltos para que uma mensagem seja recebida pelo nó sensor não é objeto de análise, uma vez que o simulador já implementa o módulo de atraso na rede e portanto não sendo determinante para a precisão do sincronismo.

6.1 Geração do Vetor de Perda

Para a geração do vetor de perda, a ser utilizado como parâmetro de entrada no componente *Message Loss* do simulador, foram realizadas simulações considerando uma aplicação de sensores em área agrícola, como por exemplo monitoramento de plantações através da utilização de drones. Com o objetivo de simular um cenário

mais próximo da situação real, foi utilizada uma área de sensoriamento circular com área equivalente a uma fazenda de 4000 hectares de área total (EMBRAPA, 2018), sensores com alcance de transmissão de 100 metros e o nó *sink* posicionado no centro da Área. A Tabela 5 apresenta os parâmetros de entrada utilizados na execução do script.

Tabela 5 - Parâmetros de Simulação – Vetor de Perda

Parâmetro	Valor
<i>raio_area - R</i>	3570 metros (m)
<i>raio_alcance - r</i>	100 metros (m)
<i>qtd_sensores</i>	Variação entre 4000 e 10000 sensores
<i>A</i>	1 (todos os nós estão ativos)
<i>qtd_execuções</i>	200

Fonte: Elaborada pelo Autor

A partir dos parâmetros configurados foram obtidos os resultados para as taxas médias de perda em função da quantidade de nós sensores, apresentados na Figura 42. Os valores gerados para cada execução serão utilizados para compor o vetor de perda que servirá como dado de entrada no simulador para a análise de sincronismo na rede. O vetor de perda gerado para cada quantidade de nós sensores é apresentado por completo no APÊNDICE D – VETORES DE PERDA

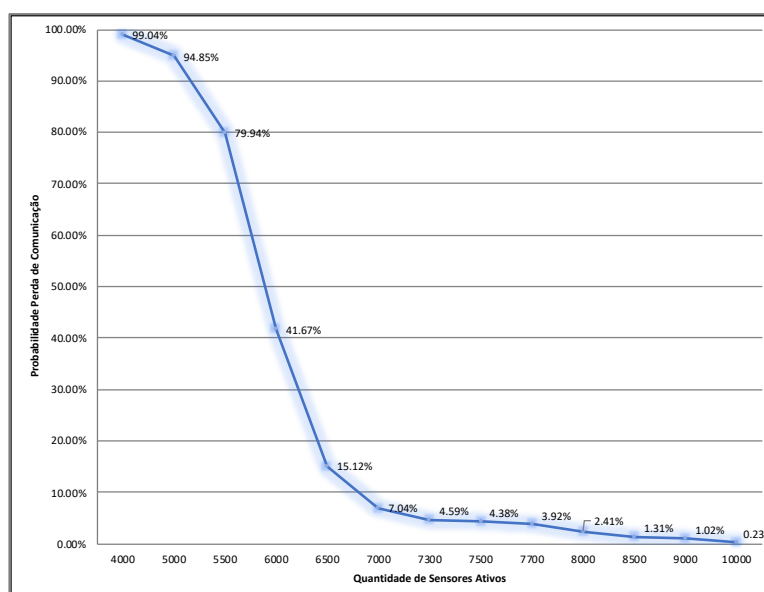


Figura 42 - Taxa Média de Perda x Quantidade de Nós Sensores

Fonte: Elaborada pelo Autor

Os resultados obtidos demonstram que uma vez que uma determinada área a ser monitorada possua uma distribuição de 4000 sensores a probabilidade de um novo sensor distribuído de forma aleatória nesta mesma área não receber mensagens a partir do nó sensor sink é de 99,04%. Porém se nesta mesma área tiverem 7000 sensores a probabilidade de perda de comunicação cai para 7,04%.

Isto significa que para um crescimento de 75% na quantidade de sensores a probabilidade de uma mensagem ser perdida por um nó sensor qualquer (X_i) é 14,07 vezes menor. Enquanto que para ter uma probabilidade de perda praticamente nula, no valor de 0,23%, é preciso aumentar o número de sensores em 2,5 vezes.

Tendo como base os dados obtidos com 8000 sensores a taxa de perda é de 2,41%. Porém para um nó sensor X_i receber uma mensagem, em um determinado instante de tempo t com uma taxa de perda de aproximadamente 0,23% é necessário aumentar em 25% a quantidade de sensores. Isto demonstra que a partir de uma determinada quantidade de nós sensores distribuídos existe uma saturação do sistema o que demonstra que o ganho de uma distribuição de nós sensores sem avaliação de cobertura é mínimo e, portanto, não é possível aumentar a precisão de sincronismo deste mesmo nó sensor.

6.2 Análise da Relação Raio de Alcance x Raio da Área de Sensoriamento

Com o objetivo de avaliar o comportamento da taxa de perda em relação a variação do raio de cobertura do sensor (r/R) e foi realizada uma simulação onde o número de nós sensores é fixo, o raio da área de sensoriamento (R) é fixo e o alcance de transmissão do nó sensor (r) é variado em passos de 50 metros, tendo como valor inicial 100m, valor este definido na simulação desenvolvida na seção 6.1 Geração do Vetor de Perda.

O número de nós foi fixado com base em resultados obtidos a partir de simulação realizada em que o raio de transmissão do nó sensor foi fixado em 0,1 do raio da área de Sensoriamento, isto é, $r = 0,1 R$ e variada a quantidade de nós sensores ativos a partir de 100 sensores com acréscimos de 100 unidades. Os resultados obtidos são apresentados na Figura 43.

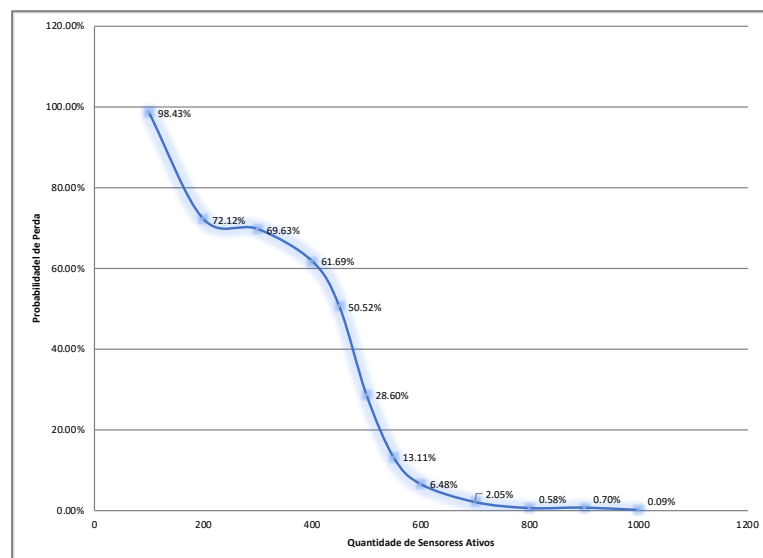


Figura 43 - Quantidade de Sensores Ativos x Taxa média de Perda – $r=0,1R$

Fonte: Elaborada pelo Autor

A partir deste resultado verificou-se que com 450 sensores ativos a taxa de perda é de 50,52% (aproximadamente 50%) e, portanto, esta foi a quantidade de nós sensores fixada para verificar o comportamento da taxa de perda com base na variação da relação r/R .

A Tabela 6 apresenta os parâmetros utilizados para a realização da simulação para análise do comportamento da perda em função da relação r/R .

Tabela 6 - Parâmetros de Simulação - Relação r/R

Parâmetro	Valor
<i>raio_area - R</i>	3570 metros (m)
<i>raio_alcance - r</i>	Variável de 100 a 750 metros em passos de 50m
<i>qtd_sensores</i>	450 sensores
<i>A</i>	1 (todos os nós estão ativos)
<i>qtd_execuções</i>	200

Fonte: Elaborada pelo Autor

O comportamento da taxa de perda em função da relação (r/R) é apresentado na Figura 44.

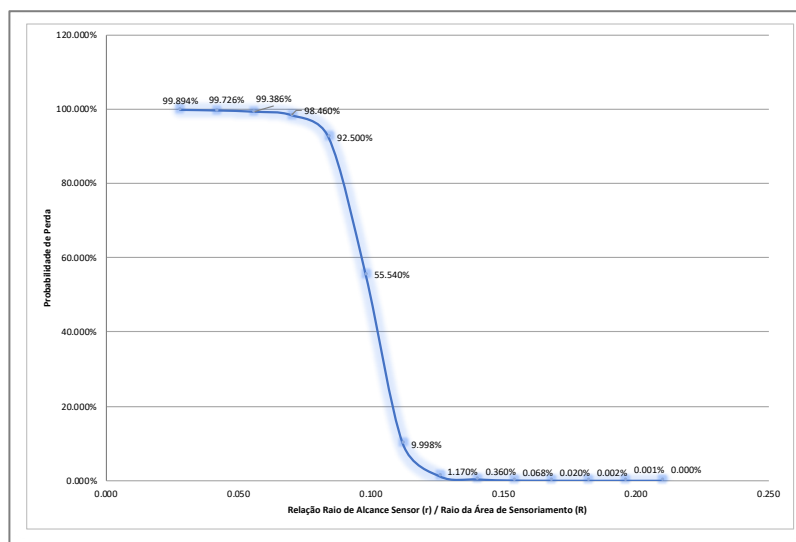


Figura 44 – Taxa Média de Perda em Função de r/R

Fonte: Elaborada pelo Autor

A análise dos resultados demonstra que a relação r/R é determinante apenas em uma faixa desta relação. A partir do valor aproximado de $r/R = 0.12$, a taxa de perda praticamente se mantém a mesma. Isto é relevante do ponto de vista de projetistas de rede de sensores que necessitam especificar equipamentos que possuam ao mesmo tempo boa cobertura, o que implica em alcances de transmissão maiores, e sensores que possuam baterias com longa duração.

6.3 Simulação Precisão de Sincronismo

Para a análise da precisão de sincronismo as simulações realizadas tiveram como dado de entrada para o vetor de perda os dados resultantes das simulações anteriores.

Para estas simulações também foram calculados os parâmetros MTIE e TDEV, uma vez que ambos são parâmetros normativos e são utilizados para determinar o erro máximo de sincronismo entre dois relógios e a estabilidade deste sincronismo.

As simulações consideram uma área de sensoriamento circular de raio R , sensores com raio de alcance de transmissão r , mobilidade dos nós sensores em alta velocidade e o nó sensor sink é fixo no centro da área de sensoriamento. Os parâmetros fixados para todas as simulações realizadas para análise de precisão de

sincronismo estão apresentados na Tabela 7, e variações específicas de algum destes parâmetros são descritas no respectivo cenário.

Tabela 7 - Parâmetros de Simulação - Precisão de Sincronismo

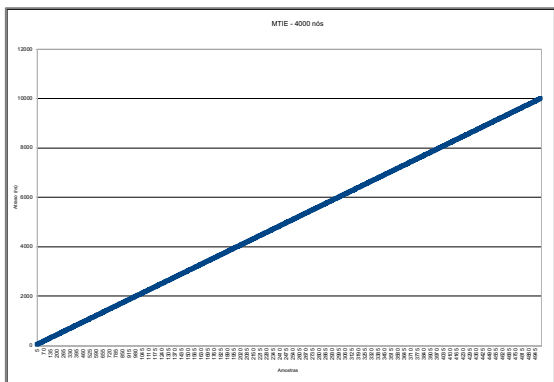
Parâmetro	Valor
<i>Vetor de Perda</i>	Obtido a partir da simulação da <u>seção 6.1 Geração do Vetor de Perda</u>
<i>Frequência Oscilador Slave</i>	99.8% da Frequência do Oscilador Master
<i>Velocidade de Deslocamento</i>	Alta

Fonte: Elaborada pelo Autor

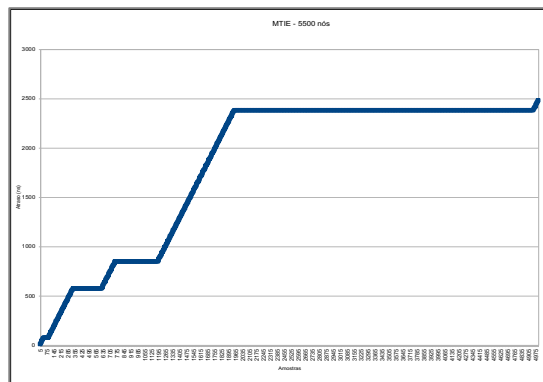
6.3.1 Cenário 1 – Precisão do Sincronismo em Função da Quantidade de Nós Sensores

O objetivo deste cenário de simulação é analisar o comportamento do sincronismo em um determinado no sensor – X_i em relação ao nó sensor sink, que tem como função ser o relógio de referência para a rede toda. Para este cenário é realizada uma simulação para cada vetor de perda obtido na simulação realizada na seção 6.1 Geração do Vetor de Perda, portanto foram gerados os gráficos de MTIE para as seguintes quantidades de nós sensores: 4000; 5000; 5500; 6000; 6500; 7000; 7300; 7500; 7700; 8000; 8500; 9000; 10000.

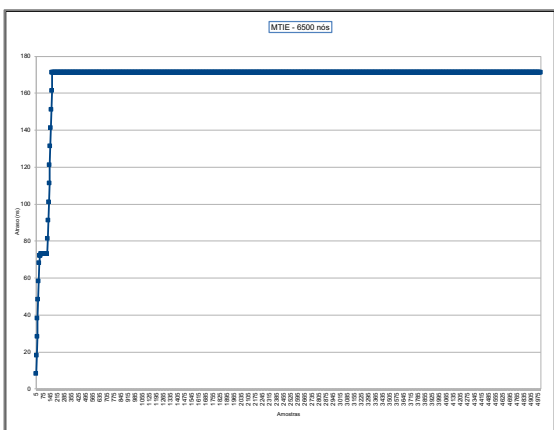
Os gráficos de MTIE para as simulações com 4000, 5500, 6500, 7500, 8000 e 10000 nós sensores, os quais abrangem o intervalo de perda de comunicação entre 99.04% a 0.23% de perda, são apresentados na Figura 45 (a até f respectivamente). Os gráficos para as demais quantidades de nós são apresentados no APÊNDICE E – GRÁFICOS MTIE.



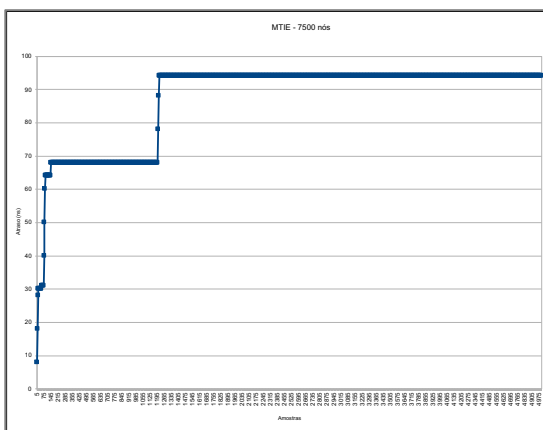
(a)



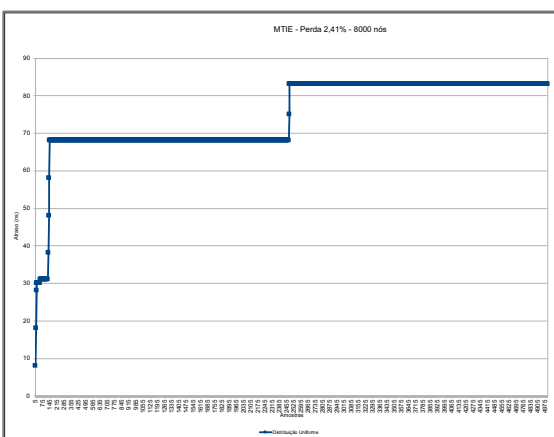
(b)



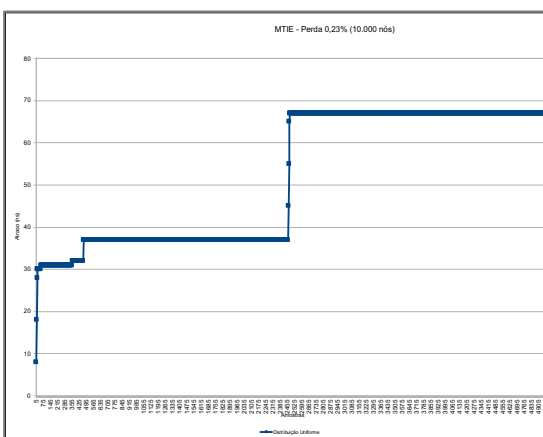
(c)



(d)



(e)



(f)

Figura 45 - Gráfico MTIE
 Fonte: Elaborada pelo Autor

Os gráficos apresentados na Figura 45 apresentam o erro máximo para as quantidades de nós sensores ativos. Como a taxa de perda de comunicação é dada em função da quantidade de nós sensores, o MTIE também pode ser interpretado como sendo em função desta mesma taxa de perda.

No gráfico (a) da Figura 45 o MTIE é crescente a uma taxa constante devido a taxa de perda ser de 99,04%, o que significa que praticamente todas as mensagens enviada pelo relógio de referência *Master_CLK* são perdidas e portanto o *Slave_CLK* não consegue efetuar a correção de sincronismo e passa a manter seu relógio com base em seu oscilador interno. A precisão de sincronismo do sistema, passa então a ser em função da precisão interna de cada oscilador.

A medida que a taxa de perda de comunicação diminui o MTIE passa a ter o comportamento de acordo com o definido nas normas ITU-T G.8261 (ITU-T G.8261, 2008), isto é passa a ter um comportamento degrau, onde por determinados intervalos de tempo o erro se mantém constante. Como pode ser observado nos demais gráficos da Figura 45, este intervalo de tempo, em que o erro se mantém constante, aumenta e o MTIE diminui, chegando um valor de 67 ns para uma taxa de perda de 0,23%, gráfico (f) da Figura 45, o que pode ser considerado como o melhor MTIE que o sistema pode entregar. Este erro máximo já demonstra a viabilidade de se manter uma rede de sensores com sincronismo baseado em troca de mensagens

6.3.2 Cenário 2 – Precisão do Sincronismo em Função da Quantidade de Nós Sensores – Taxa de Perda de 5%

Considerando que as redes de sensores quando utilizadas para aplicações em tempo real, como por exemplo de voo de drones em aplicações de transporte, pode ter a segurança comprometida devido a perda de informação de sincronismo, esta simulação tem por objetivo analisar mais especificamente o intervalo de perda que pode comprometer este tipo de aplicação.

Para definir o intervalo de análise, uma vez que as simulações realizadas no cenário 1 abrangem o intervalo com taxa de perda de comunicação entre 99.04% e 0.23%, foi utilizado como base aplicações VOIP, onde a taxa de perdas de pacotes aceitável é

de 1% e para perdas de pacotes acima de 5% do total, uma conversa VoIP fica comprometida (ROSS, 2007).

Os parâmetros utilizados para as simulações deste cenário estão informados na Tabela 8.

Tabela 8 - Parâmetros de Simulação - Precisão de Sincronismo – Cenário 2

Parâmetro	Valor
<i>Vetor de Perda</i>	Obtido a partir da simulação da <u>seção 6.1 Geração do Vetor de Perda</u> para: 7000; 7300; 7500; 7700 e 8000 nós sensores
<i>Frequência Oscilador Slave</i>	99.8% da Frequência do Oscilador Master
<i>Velocidade de Deslocamento</i>	Alta

Fonte: Elaborada pelo Autor

A Figura 46 apresenta o gráfico com a comparação dos valores de MTIE para as quantidades de nós sensores descritos na Tabela 8, de forma comparativa.

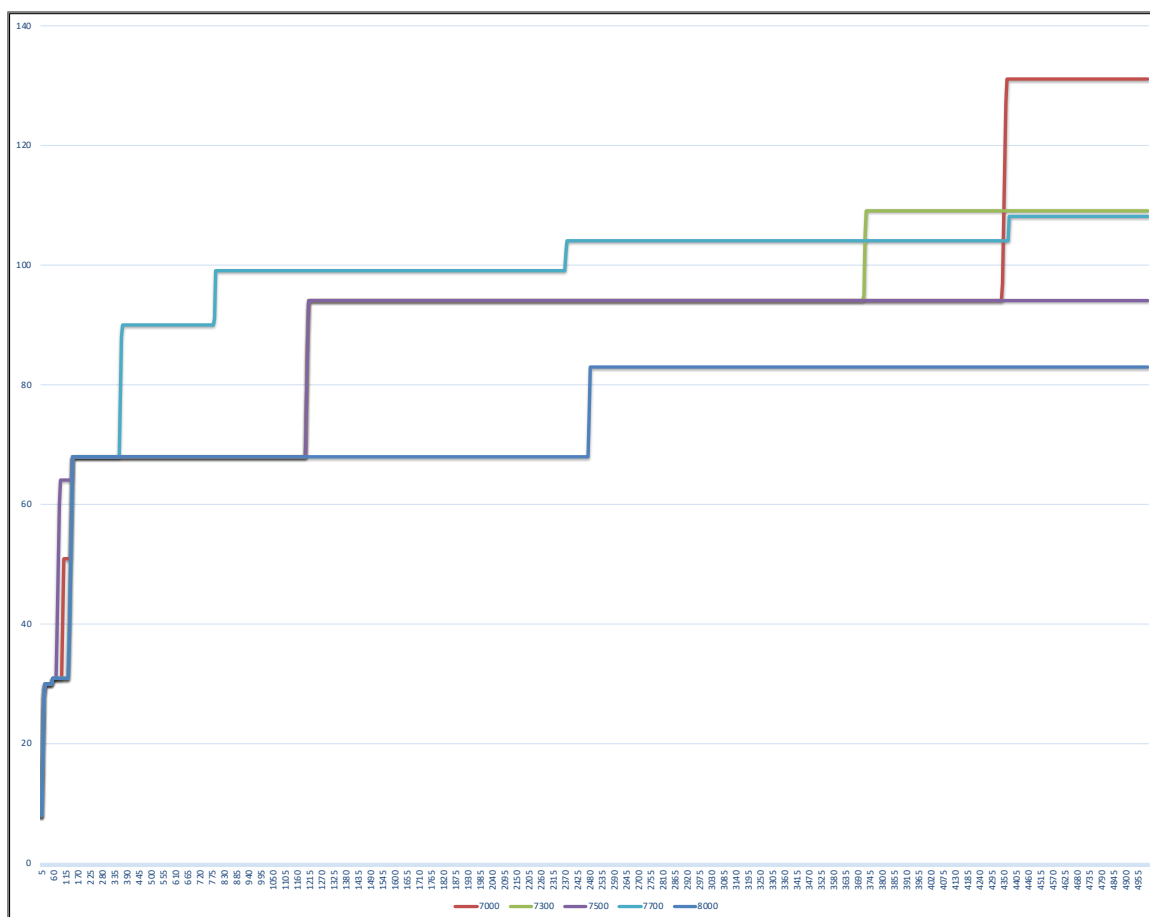


Figura 46 - Parâmetro MTIE - Intervalo de Taxa de Perda [7,04% - 2,41%]

Fonte: Elaborada pelo Autor

Como pode ser observado o MTIE para intervalo de perda analisado se matem com no máximo 131ns para 7000 nós sensores ou 7,04% de taxa de perda e se mantém no menor valor, 83ns para 8000 nós. Apesar da diferença entre o maior valor e o menor valor ser de 58%, ainda é possível manter a precisão de sincronismo na ordem de ns, gerando resultados melhores que a utilização de protocolos como NTP. O impacto gerado pela maior perda se dá nos intervalos onde o MTIE cresce, ou seja para perdas maiores, mesmo se mantendo em ns, o MTIE cresce de maneira mais rápida que para perdas menores, o que pode comprometer o sincronismo da rede quando se mantém o sistema em utilização por grandes intervalos de tempo (dias).

6.3.2.1 Análise da Estabilidade do Sincronismo

Apesar do MTIE ser um parâmetro normativo, não é possível avaliar a estabilidade do sincronismo somente através de seu cálculo. Portanto para analisar esta característica foi realizado o cálculo do parâmetro TDEV. Como o objetivo do TDEV não é analisar erro e sim sua estabilidade, ou seja, o quanto a diferença entre os relógios *MASTER_CLK* e *SLAVE_CLK* se mantém ao longo do tempo, foram considerados os vetores de perda e o MTIE máximo para 5000, 7300 e 9000 nós sensores. A Tabela 9 apresenta os valores de MTIE e perda para estas quantidades de sensores.

Tabela 9 - MTIE e Perda - Cálculo TDEV

Quantidade de Nós Sensores	MTIE (ns)	Taxa de Perda
5000	6000	94,85%
7300	109	4,59%
9000	83	1,02%

Fonte: Elaborada pelo Autor

A Figura 47 apresenta o gráfico com o parâmetro TDEV obtido a partir dos resultados das simulações.

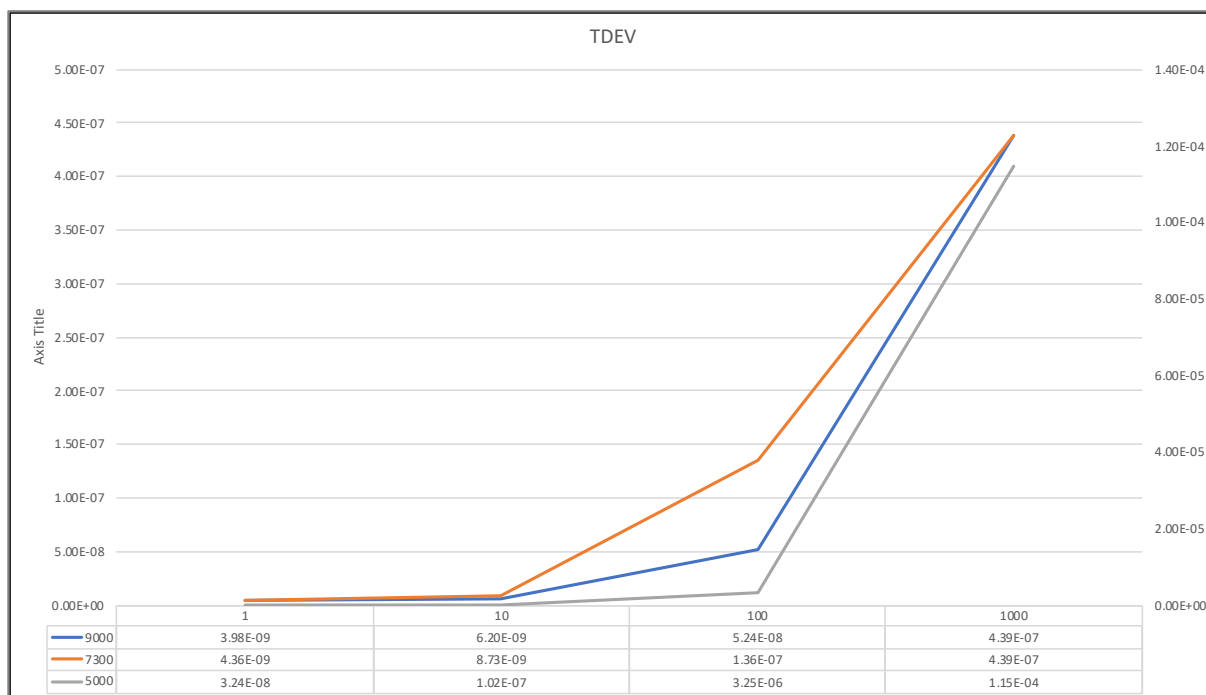
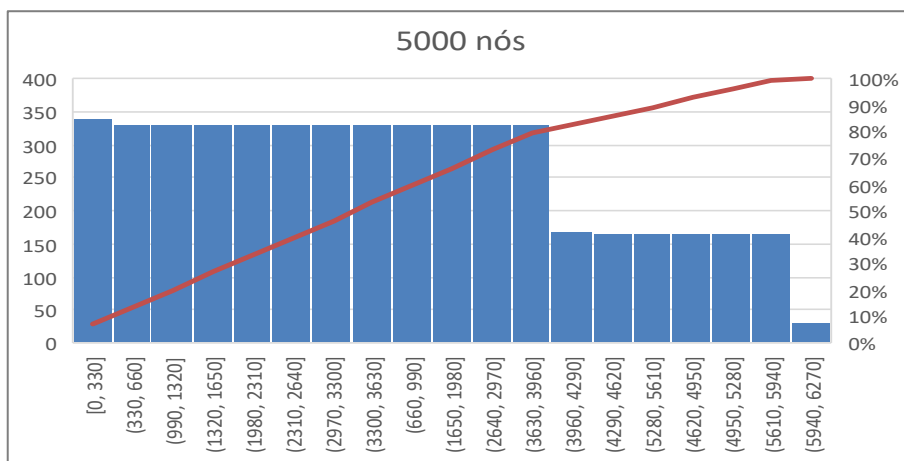


Figura 47 - Parâmetro TDEV

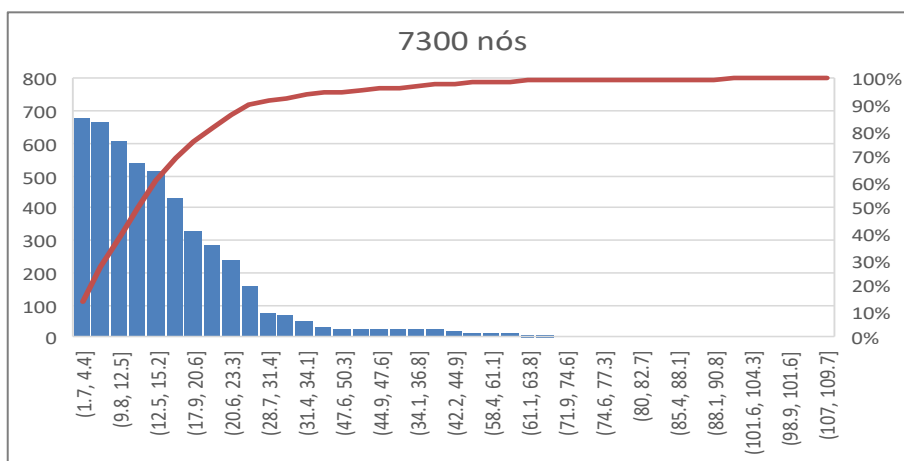
Fonte: Elaborada pelo Autor

O cálculo do parâmetro TDEV tem o comportamento adequado conforme descrito nas normas ITU-T e para os três valores de perda analisados é possível concluir que o sistema se mantém estável. A diferença é que para perdas maiores o deslocamento entre o *MASTER_CLK* e o *SLAVE_CLK* é maior, o que pode inviabilizar aplicações em tempo real. Outra diferença entre os valores de TDEV obtidos é o intervalo de tempo em que o sincronismo se mantém estável. Enquanto para intervalos de tempo de simulação da ordem de 10^3 obtém-se estabilidade da ordem de 10^{-7} para 9000 nós sensores, a estabilidade cai para 10^{-4} para 5000 nós sensores.

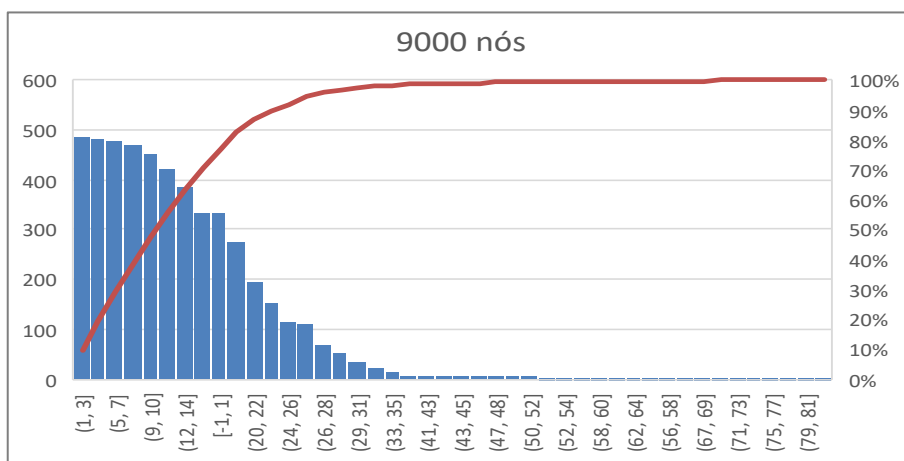
Os gráficos (a), (b) e (c) apresentados na Figura 48 apresentam os histogramas para as mesmas quantidade de nós sensores analisados no parâmetro TDEV de forma a se obter uma visão mais completa do comportamento do sistema de forma a viabilizar a correlação entre estabilidade de sincronismo, perda de mensagens e atraso.



(a)



(b)



(c)

Figura 48 – Histogramas
 Fonte: Elaborada pelo Autor

Pelos histogramas é possível analisar que apesar do TDEV para 5000 nós sensores demonstrar estabilidade 90% das mensagens transmitidas apresentam um atraso da ordem de 5000 ns, enquanto que para 9000 sensores 90% das mensagens apresentam atraso máximo de 28 ns, mesmo o MTIE sendo de 83ns, isto é, 66% menor que o MTIE para o mesmo intervalo de tempo. Já para 7300 nós sensores, equivalente a uma taxa de perda de 4,59%, o MTIE é de 109 ns, enquanto que para 90% das mensagens recebidas e processadas pelo nó sensor *SLAVE_CLK* é de 23.3ns ou seja 78,89% menor que o MTIE.

Esta análise conjunta dos parâmetros MTIE, TDEV e histogramas demonstra a capacidade do sistema em se manter sincronizado com precisão de ns, mesmo quando existem perdas significativas que podem comprometer a qualidade e a segurança na utilização deste tipo de rede para aplicações de tempo real.

6.3.3 Cenário 3 – Variação da Frequência Master / Slave

Esta simulação tem como objetivo analisar como o sistema consegue manter a correção quando a diferença de estabilidade de frequência dos osciladores varia. Este tipo de análise é importante pois os fabricantes de osciladores garantem a precisão de sincronismo dentro de uma faixa de precisão de $\pm 2\%$ e quando o sistema perde a mensagem de sincronismo enviada pelo *MASTER_CLK* ou a mensagem enviada sofre atraso, o mecanismo de correção do *SLAVE_CLK* se mantém baseado em seu oscilador interno.

Para esta análise foi fixada a quantidade de nós sensores em 7300, o que resulta em uma taxa média de perda de comunicação de 4,59%. Como o objetivo é analisar como o atraso se comporta sob estas condições, o parâmetro analisado é o MTIE.

A variação de frequência foi estabelecida a partir da fixação da frequência do oscilador interno do *MASTER_CLK* e foi variada a frequência interna do oscilador do *SLAVE_CLK*. Esta variação foi estabelecida de acordo com o indicado na Tabela 10.

Tabela 10 - Diferença de Frequência Master / Slave

Quantidade de Nós Sensores	Frequência Slave
7300	Slave Atrasa 1% em relação ao Master
7300	Slave Atrasa 0,5% em relação ao Master
7300	Slave Atrasa 0,2% em relação ao Master *
7300	Slave igual ao Master
7300	Slave Adianta 0,5% em relação ao Master
7300	Slave Adianta 1% em relação ao Master

Fonte Elaborada pelo Autor

*Diferença de frequência adotada nos cenários 1 e 2 de simulação

A Figura 49 apresenta o gráfico comparativo dos valores de MTIE para as simulações realizadas.

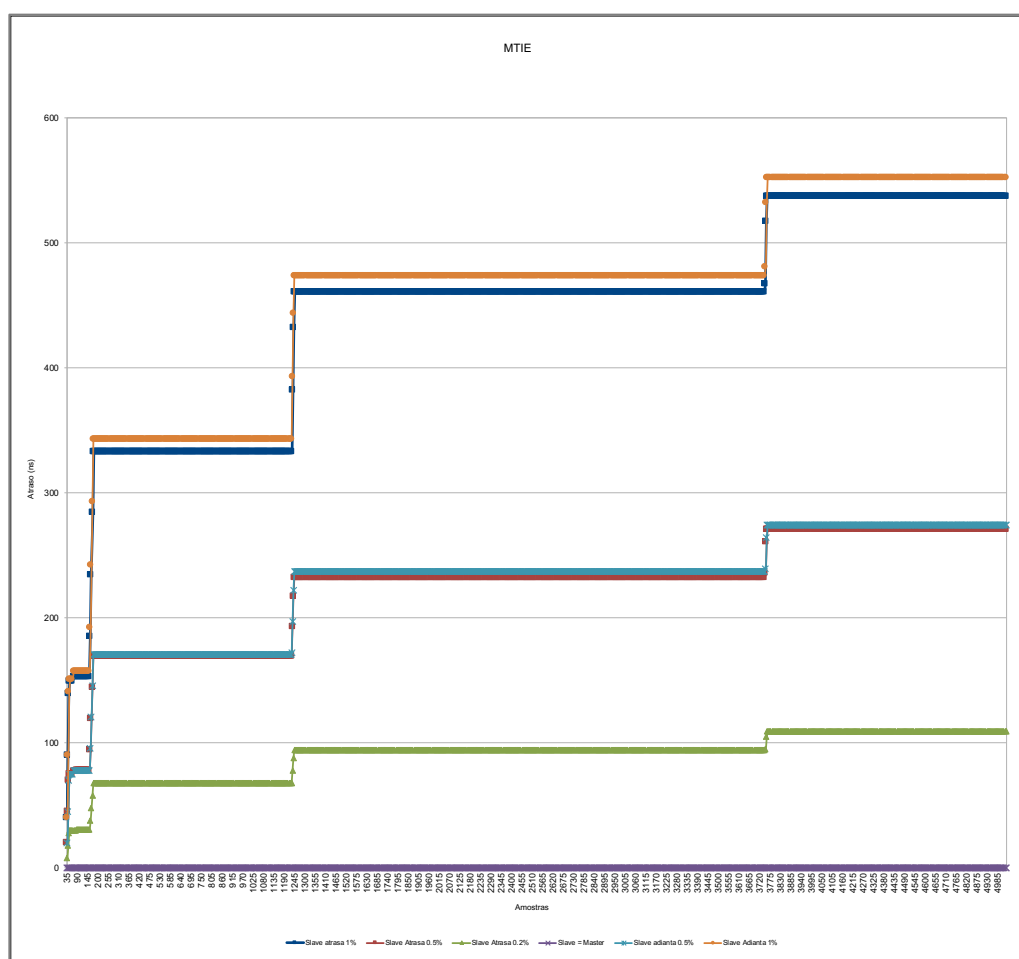


Figura 49 - MTIE Diferença de Frequência

Fonte: Elaborada pelo Autor

Os resultados apresentados demonstram a capacidade do sistema manter a correção em ns mesmo quando ocorre a variação de frequência do SLAVE_CLK. O fator de atrasar ou adiantar não é decisivo para o aumento do MTIE. Este parâmetro é diretamente correlacionado com a imprecisão da frequência do oscilador. Porém mesmo para diferenças de 1% de variação o sistema se manteve sincronizado com precisão de ns, tendo um erro máximo de 552 ns.

7 CONSIDERAÇÕES FINAIS

Este trabalho teve como tema central a precisão de sincronismo em redes de sensores e tem sido estimulado pelos avanços tecnológicos e pelas novas aplicações baseadas em sensores, como por exemplo a aplicação em FANETS, que traz novas necessidades devido ao ambiente dinâmico gerado por este tipo de rede.

O acesso a este tipo de informação antecipadamente é de fundamental importância, pois permite que projetistas de redes de sensores e desenvolvedores de aplicações possam avaliar o comportamento da rede e assim agir previamente, modificando seus projetos ou até mesmo limitando cenários de uso, principalmente para aplicações críticas onde requisitos como precisão de sincronismo quando não atendidos podem impactar diretamente na segurança dos usuários.

Este capítulo finaliza o trabalho, descrevendo as suas principais contribuições relativas ao tema estudado, e apresentando propostas de trabalhos futuros relevantes que podem ser realizados.

7.1 Contribuições

Os resultados quantitativos obtidos permitem validar a proposição de que um sistema de sincronização baseado na transmissão de marcas de tempo (*timestamps*) em pacotes de um protocolo e o ajuste do relógio do nó escravo em função destas marcas de tempo pode ser utilizado para a realização de sincronização de redes de sensores.

Esta possibilidade é mantida mesmo que o trânsito de pacotes na rede apresente atrasos variáveis e estejam sujeitos a perdas de mensagens. O modelo implementado faz uma aproximação da taxa de envio de mensagens, do atraso da rede através de distribuições probabilísticas e da taxa de perda baseada na quantidade de nós sensores ativos, área de sensoriamento, alcance de transmissão do nós sensor e mobilidade.

Para os cenários estudados em que foi considerada mobilidade com alta velocidade dos nós, aderente a aplicações de DRONES em FANETs, a variação máxima de pico-a-pico de atraso obtida foi da ordem de nano-segundos mantendo-se aproximadamente 90% das mensagens transmitidas com atraso inferior ao máximo.

Desta forma as contribuições deste trabalho podem ser resumidas como:

- a) A possibilidade de utilização de protocolos de sincronismo baseados em mensagens em ambientes que exijam alta precisão;
- b) Sob o aspecto das variáveis analisadas este trabalho contribui para que aplicações com requisitos críticos, principalmente de sincronismo, possam ter seu ambiente de rede projetados e analisados de forma antecipada, evitando assim a ocorrência de situações de insegurança.
- c) Através do simulador implementado é possível analisar o comportamento de um determinado nó sensor em situações específicas de perda ou atraso, facilitando a tomada de decisão de operadores que utilizem este tipo de rede para aplicações de transporte, garantindo assim a segurança de todo o sistema e de seus usuários.
- d) A construção de um simulador baseado em probabilidades que não depende de dados reais dos elementos que compõem a rede para ter seus comportamentos avaliados
- e) Script de geração de taxa de perda baseado na área de sensoriamento, alcance de transmissão, quantidade de sensores, permitindo assim que projetistas de redes possam de forma simples obter a taxa de perda associada ao ambiente projetado e assim avaliar o atendimento aos requisitos da aplicação.
- f) A disseminação do conhecimento científico adquirido por meio de publicações em congressos e periódicos dos trabalhos intitulados “*A Simulation Tool for Synchronism Distribution Based on Standard IEEE 1588*” (EIRAS; ZUCCHI, 2014) “*Development of an IEEE 1588 Simulator and Analysis of UAVs’ Synchronization in a FANET Environment*” (EIRAS; ZUCCHI, 2015), “*A Statistical Method for Area Coverage Estimation and Loss Probability Analysis on Mobile Sensor Networks*” (EIRAS; ZUCCHI, 2019). “*A simulation Model for Area Convergence and Loss Probability on Mobile Sensor Networks*” e “*Measuring Synchronization Precision in Mobile Sensor Networks*”.

7.2 Trabalhos futuros

Entre as possibilidades futuras para estudos na área e evolução deste trabalho, vale mencionar:

- a) a implementação do modelo de baixa velocidade no simulador desenvolvido, gerando assim a possibilidade de aplicação deste mesmo estudo para outros tipos de aplicação, como por exemplo avaliação de microclima para agricultura de precisão.
- b) Avaliação de implementação de mecanismos de clusterização de sincronismo com a utilização de relógios de hierarquia intermediária, permitindo assim que mesmo que haja alta taxa de perdas de mensagens os nós sensores consigam manter o sincronismo na ordem de nanosegundos.
- c) Implementação de uma rede de sensores baseada por exemplo em DRONES reais, para análises de outras variáveis ambientais que possam influenciar na perda ou atraso das mensagens de sincronismo.
- d) Implementação de um componente no simulador que possa variar a precisão de frequência dos osciladores durante a execução da simulação, pois em situações reais a precisão do oscilador interno dos relógios pode variar de acordo com temperatura e bateria entre outras variáveis.
- e) Levantamento e análise de todas as variáveis de contexto que impactam uma rede de sensores móveis e onde esta rede está implementada com o objetivo de desenvolver algoritmos de inteligência artificial que possam controlar em tempo real os estados dos sensores e suas variáveis como velocidade de deslocamento e prioridade de transmissão, para que a rede mantenha os requisitos de qualidade de serviço demandados pela aplicação, mantendo seus usuários em segurança.

REFERÊNCIAS

ALEKSANDER, N. et al. Differential navigation for UAV platforms with mobile reference station. In: FUJITA, H.; SASAKI, J. (Eds.). . **Selected Topics in Applied Computer Science**. Japan: WSEAS Press, 2010. p. 465–471.

ALIA, O. M.; AL-AJOURI, A. Maximizing Wireless Sensor Network Coverage With Minimum Cost Using Harmony Search Algorithm. **IEEE Sensors Journal**, v. 17, n. 3, p. 882–896, 2017.

ANYAEGBU, M.; WANG, C.; BERRIE, W. Dealing with Packet Delay Variation in IEEE 1588 Synchronization Using a Sample-Mode Filter. **IEEE Intelligent Transportation Systems Magazine**, p. 20–27, 2013.

ASTUDILLO, G.; KADOCH, M. A Location-aware Routing Scheme for Wireless Mesh Networks using Directional Antennas. In: MUSIC, J. (Ed.). . **Recent Advances in Computer, Communications and Information Technology**. Tenerife: WSEAS Press, 2014. p. 57–62.

AURENHAMMER, F. Voronoi diagrams - A Survey of a Fundamental Geometric Data Structure. **ACM Computing Surveys (CSUR)**, v. 23, n. 3, p. 345–405, 1991.

AZEVEDO, A. L. **Dano ambiental em Brumadinho ameaça centenas de espécies**. Disponível em: <<https://oglobo.globo.com/brasil/dano-ambiental-em-brumadinho-ameaca-centenas-de-especies-23424033>>. Acesso em: 8 jul. 2019.

AZIZ, N. A. B. A.; MOHEMMED, A. W.; ALIAS, M. Y. **A Wireless Sensor Network Coverage Optimization Algorithm Based on Particle Swarm Optimization and Voronoi Diagram**. 2009 International Conference on Networking, Sensing and Control. **Anais...Okayama, Japan: IEEE, 2009** Disponível em: <<http://ieeexplore.ieee.org/document/4919346/>>

BEKMEZCI, İ.; SAHINGOZ, O. K.; TEMEL, Ş. Flying Ad-Hoc Networks (FANETs): A survey. **Ad Hoc Networks**, v. 11, n. 3, p. 1254–1270, 2013a.

BEKMEZCI, İ.; SAHINGOZ, O. K.; TEMEL, Ş. Flying Ad-Hoc Networks (FANETs): A survey. **Ad Hoc Networks**, v. 11, n. 3, p. 1254–1270, maio 2013b.

CHO, H. C. H. et al. Precision Time Synchronization Using IEEE 1588 for Wireless Sensor Networks. **2009 International Conference on Computational Science and Engineering**, v. 2, 2009.

CONTI, M.; GIORDANO, S. Mobile ad hoc networking: Milestones, challenges, and new research directions. **IEEE Communications Magazine**, v. 52, n. 1, p. 85–96, 2014.

DECEA. ICA 100-40 - Sistemas de Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro. p. 76, 2015.

DEIF, D. S.; GADALLAH, Y. Classification of Wireless Sensor Networks Deployment Techniques. **IEEE Communications Surveys & Tutorials**, v. 16, n. 2, p. 834–855, 2014.

DEL RÍO, J. et al. Precision timing in ocean sensor systems. **Measurement Science and Technology**, v. 23, n. 2, p. 025801, 2012.

DOBROGOWSKI, A.; KASZNIA, M. Methods of Real-time Calculation of Allan Deviation and Time Deviation. v. 1, n. 2, p. 42–46, 2010.

EIRAS, F. C. DA S.; ZUCCHI, W. L. **A Simulation Tool for Synchronism Distribution Based on Standard IEEE 1588**. Recent Advances in Electrical and Electronic Engineering. **Anais...Florence: 2014**

EIRAS, F. C. DA S.; ZUCCHI, W. L. Development of an IEEE 1588 Simulator and Analysis of UAVs ' Synchronization in a FANET environment. **International Journal of Circuits, Systems and Signal Processing**, v. 9, p. 94–102, 2015.

EIRAS, F. C. DA S.; ZUCCHI, W. L. **A Statistical Method for Area Coverage Estimation and Loss Probability Analysis on Mobile Sensor**. 6th International Electronic Conference on Sensors and Applications. **Anais...2019**

ELSON, J.; GIROD, L.; ESTRIN, D. Fine-grained network time synchronization using reference broadcasts. **ACM SIGOPS Operating Systems Review**, v. 36, n. SI, p. 147, 17 maio 2002.

ELY, T. A. et al. Using the Deep Space Atomic Clock for Navigation and Science. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 65, n. 6, p. 950–961, 2018.

EMBRAPA. **EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária**. Disponível em: <<https://www.embrapa.br/>>. Acesso em: 10 mar. 2018.

EUROCONTROL. **EUROCONTROL Specification for ATM Surveillance System Performance (Volume 1)**. Brussels: EUROCONTROL Headquarters, 2012. v. 1

FEDERAL AVIATION ADMINISTRATION. **Summary of Small Unmanned Aircraft Rule (Part 107)**. Washington. DC: [s.n.]. Disponível em: <http://www.faa.gov/uas/media/Part_107_Summary.pdf>.

GANERIWAL, S.; KUMAR, R.; SRIVASTAVA, M. B. Timing-sync protocol for sensor networks. **Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03**, p. 138–149, 2003.

GUPTA, N.; KUMAR, N.; JAIN, S. **Coverage Problem in Wireless Sensor Networks: a Survey**. International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016. **Anais...2016**

HATHAWAY, R. J.; BEZDEK, J. C. NERF c-means: Non-Euclidean Relational Fuzzy Clustering. **Pattern Recognition**, v. 27, n. 3, p. 429–437, 1994.

HE, S. et al. Cost-effective barrier coverage by mobile sensor networks. **Proceedings - IEEE INFOCOM**, p. 819–827, 2012.

HIRANI, P. K. (JAYOTI V. W. U.; SINGH, M. ((JAYOTI V. W. U. A Survey on Coverage Problem in Wireless Sensor Network. **International Journal of Computer Applications**, v. 116, 2015.

HOLLAND, J. H. Genetic Algorithms and Adaptation. **Scientific American**, p. 66–72, jul. 1992.

ICAO. **Terminal Navigation Specifications - RNAV 2, RNAV 1 and Basic RNP-1**. (ICAO, Ed.) ICAO PBN Seminar. **Anais...Montreal: 2007**

IEEE. **IEEE Std 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems**. [s.l: s.n.]. v. 2008

IEEE. **IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**. New York, New York, USA: IEEE, 2012. v. 2012

ITU-T G.810. **ITU-T G.810 Definitions and terminology for synchronization networks**, 1996.

ITU-T G.811.1. **ITU-T G811.1 - Timing characteristics of primary reference clocks**, 2016.

ITU-T G.812. **ITU-T G.812 - Timing requirements of slave clocks suitable for use as node clocks in synchronization networks**, 2004.

ITU-T G.813. **ITU-T G.813 - Timing characteristics of SDH equipment slave clocks (SEC)**, 2003.

ITU-T G.8261. **ITU-T G.8261/Y.1361 Timing and Synchronization Aspects in Packet Networks**, 2008.

ITU-T G.8262. **ITU-T G.8262/Y.1362 - Timing Characteristics of a Synchronous Ethernet Equipment Slave Clock**, 2010.

KRONENBERG, J. L. M. **Contribuições para o Fortalecimento do Laboratório Brasileiro de Metrologia de Tempo e Freqüência**. [s.l.] Pontifícia Universidade Católica do Rio de Janeiro, 2007.

KULKARNI, R. V; MEMBER, S.; KUMAR, G. Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey. **IEEE Transactions on Systems, Man, and Cybernetics - PART C: Applications and Reviews**, v. 41, n. 2, p. 262–267, 2011.

LIU, Y.; YANG, C. OMNeT++ based modeling and simulation of the IEEE 1588 PTP clock. **2011 International Conference on Electrical and Control Engineering**, p. 4602–4605, set. 2011.

MARÓTI, M. et al. **The Flooding Time Synchronization Protocol**. ACM International Conference on Embedded Networked Sensor Systems. **Anais...2004**

MARTIN, J.; KASCH, W. **RFC 5905 - Network Time Protocol V4 - Protocol and Algorithms Specification**, 2010.

MATHWORKS. **SIMULINK Academic Version, R2011**, 2011.

MILLS, D. L. Internet Time Synchronization: The Network Time Protocol. **IEEE Transactions on Communications**, v. 39, n. 10, p. 1482–1493, 1991.

NASSER, R. B. **McCloud Service Framework : Arcabouço para desenvolvimento de serviços baseados na Simulação de Monte Carlo na Cloud**. [s.l.] Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio, 2012.

NOAA. **NOAA**. Disponível em: <<https://data.noaa.gov/dataset/noaa-next-generation-radar-nexrad-level-iii-products>>. Acesso em: 7 abr. 2017.

NOVICK, A. N.; LOMBARDI, M. A. Practical limitations of NTP time transfer. **2015 Joint Conference of the IEEE International Frequency Control Symposium and the European Frequency and Time Forum, FCS 2015 - Proceedings**, n. April, p. 570–574, 2015.

OFFICE OF THE SECRETARY OF DEFENSE. DoD UAV RoadMap - 2002 / 2027. 2002.

PIETILÄINEN, A. **Packet TDEV, MTIE, and MATIE - for Estimating the Frequency and Phase Stability of a Packet Slave Clock**. [s.l: s.n.].

PORTAL BRASIL. **Portal Brasil - Ciência e Tecnologia**. Disponível em: <<http://www.brasil.gov.br/ciencia-e-tecnologia/2012/11/rede-de-radares-meteorologicos-do-brasil-sera-ampliada-a-partir-de-2013>>. Acesso em: 7 abr. 2017.

RAMESH, M. V. Design, development, and deployment of a wireless sensor network

for detection of landslides. **Ad Hoc Networks**, v. 13, p. 2–18, fev. 2014.

RILEY, W. J. **Handbook of Frequency Stability Analysis**. [s.l.: s.n.]. v. 31

RILEY, W. J. **The Statistics of Time Transfer**. [s.l.: s.n.].

ROMOOZI, M. et al. Genetic Algorithm for Energy Efficient & Coverage-Preserved Positioning in Wireless Sensor Networks. **Proceedings - 2010 International Conference on Intelligent Computing and Cognitive Informatics, ICICCI 2010**, p. 22–25, 2010.

ROSS, J. **Livros Informações bibliográficas**. Rio de Janeiro: Antenna Edições Técnicas Ltda, 2007.

SAHINGOZ, O. K. Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges. **Journal of Intelligent & Robotic Systems**, v. 74, n. 1–2, p. 513–527, 2 out. 2013.

SANNA, A.; PRALIO, B. **Simulation and Control of Mini UAVs**. Proceedings of the 5th WSEAS Int Conf. on Simulation, Modeling and Optimization. **Anais...Corfu - Greece: 2005**

SENOUCI, M. R. et al. Movement-Assisted Sensor Deployment Algorithms : A Survey and Taxonomy. **IEEE Communications Surveys & Tutorials**, v. 17, n. April 2016, p. 2493–2510, 2015.

SIQUEIRA, R. H. DE. **CARACTERIZAÇÃO E CALIBRAÇÃO DE UNIDADE DE GIROS EM CONFIGURAÇÃO REDUNDANTE PARA TESTE DE DESEMPENHO EM DETERMINAÇÃO DE ATITUDE**. [s.l.] INPE - Instituto Nacional de Pesquisas Espaciais, 2014.

SOMMER, P.; WATTENHOFER, R. Gradient Clock Synchronization in Wireless Sensor Networks. **Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on**, p. 37–48, 2009.

WANG, B.; LIM, H. B.; MA, D. A survey of movement strategies for improving network coverage in wireless sensor networks. **Computer Communications**, v. 32, n. 13–14, p. 1427–1436, 2009.

WANG, Y. et al. Coverage problem with uncertain properties in wireless sensor networks: A survey. **Computer Networks**, v. 123, p. 200–232, 2017.

WOBSCHALL, D.; MA, Y. Synchronization of wireless sensor networks using a modified IEEE 1588 protocol. **2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication**, p. 67–70, set. 2010.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, v. 52, n. 12, p. 2292–2330, ago. 2008.

ZENG, Y.; ZHANG, R.; LIM, T. J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. **IEEE Communications Magazine**, v. 54, n. 5, p. 36–42, 2016.

APÊNDICE A – RESOLUÇÃO EQUAÇÃO

A probabilidade de um nó sensor X_2 se comunicar com o nó sensor sink através do nó sensor X_1 , sendo que este nó sensor X_1 se comunica diretamente com S ($X_2 \rightarrow X_1 \rightarrow S$) em função de ρ pode ser calculada conforme a equação abaixo.

$$P \{X_2 \rightarrow X_1 \rightarrow S\} = \frac{1}{\pi R^2} \int_0^r r^2 \left[\pi - 2 \arccos \frac{\rho}{2r} + \text{sen} \left(2 \arccos \frac{\rho}{2r} \right) \right] \frac{2\rho \partial \rho}{R^2} \quad (42)$$

Descondicionando a probabilidade é possível eliminar o parâmetro ρ e calcular a probabilidade em função dos parâmetros R (raio da área de sensoriamento) e r (raio de alcance de transmissão do nó sensor). A equação 42 pode ser dividida em três partes e cada parte pode ser descondicionada de forma independente. A seguir é apresentada a resolução de cada parte.

A.1 – Parte 1

$$\text{Parte 1} = \int_0^r r^2 \pi \cdot \frac{2\rho \partial \rho}{R^2}$$

$$= \frac{2 r^2 \pi}{R^2} \int_0^r \rho \partial \rho$$

$$= \frac{2 r^2 \pi}{R^2} \cdot \frac{r^2}{2}$$

$$\text{Parte 1} = \frac{r^4 \pi}{R^2}$$

A.2 – Parte 2

$$\begin{aligned}
 \text{Parte 2} &= \int_0^r r^2 \left(-2 \arccos \frac{\rho}{2r} \right) \cdot \frac{2\rho \partial\rho}{R^2} = \\
 &= \frac{4r^2}{R^2} \int_0^r \rho \arccos \frac{\rho}{2r} \partial\rho
 \end{aligned}$$

Sendo:

$$x = \frac{\rho}{2r}$$

Para $\rho \rightarrow 0$ então $x \rightarrow 0$

Para $\rho \rightarrow r$ então $x \rightarrow 1/2$

$$\int \arccos(x) \partial x = x \arccos(x) - \sqrt{1-x^2} + C$$

$$\partial\rho = 2r \partial x$$

Temos:

$$\begin{aligned}
 &\frac{-4r^2}{R^2} \int_0^{1/2} 2r x \arccos(x) 2r \partial x = \\
 &= \frac{4r^2}{R^2} \cdot 4r^2 \int_0^{1/2} x \arccos(x) \partial x = \\
 &= \frac{16r^4}{R^2} \int_0^{1/2} x \arccos(x) \partial x
 \end{aligned}$$

Sabendo que:

$$\int u \partial v = u.v - \int v \partial u$$

Temos:

$$u = \arccos(x)$$

$$\partial u = \frac{-1}{\sqrt{1-x^2}} \partial x$$

$$\frac{16 r^4}{R^2} \int_0^{1/2} x \arccos(x) \partial x$$

Fazendo

$$I = \int_0^{1/2} x \arccos(x) \partial x$$

$$I = \left[\frac{x^2}{2} \arccos(x) \Big|_0^{1/2} - \int_0^{1/2} \frac{x^2}{2} \cdot \frac{-1}{\sqrt{1-x^2}} \partial x \right] =$$

$$= \left[\frac{x^2}{2} \arccos(x) \Big|_0^{1/2} + \frac{1}{2} \int_0^{1/2} \frac{x^2}{\sqrt{1-x^2}} \cdot \partial x \right] =$$

Para calcular

$$\frac{1}{2} \int_0^{1/2} \frac{x^2}{\sqrt{1-x^2}} \cdot \partial x$$

Sabendo que:

$$x = \text{sen}(u)$$

$$u = \text{arcsen}(u)$$

$$dx = \cos(u) \partial u$$

Temos:

$$I = \frac{1}{2} \int \text{sen}^2 u \partial u + \frac{x^2}{2} \arccos(x) \Big|_0^{1/2}$$

$$\text{sen}^2 u = \frac{1}{2} - \frac{1}{2} \cos(2u)$$

$$I = \frac{1}{2} \int \frac{1}{2} \partial u - \frac{1}{2} \int \frac{\cos(2u)}{2} \partial u + \frac{x^2}{2} \arccos(x)$$

Fazendo

$$s = 2u \quad e \quad \partial s = 2 \partial u$$

Temos:

$$I = -\frac{1}{8} \int \cos(s) \partial s + \frac{1}{2} \int \frac{\partial u}{2} + \frac{x^2}{2} \arccos(x)$$

$$I = -\frac{1}{8} \text{sen}(s) + \frac{u}{4} + \frac{x^2}{2} \arccos(x)$$

Retornando s por 2u temos:

$$I = \frac{u}{4} - \frac{\operatorname{sen}(2u)}{8} + \frac{x^2}{2} \arccos(x) + C$$

Substituindo $u = \operatorname{arcsen}(x)$

Temos:

$$\operatorname{sen}(2\operatorname{arcsen}(x)) = 2 \cos(\operatorname{arcsen}(x)) \operatorname{sen}(\operatorname{arcsen}(x)) = \frac{2x}{\sqrt{1-x^2}}$$

Portanto

$$I = \frac{\operatorname{arcsen}(x)}{4} - \frac{x}{4\sqrt{1-x^2}} + \frac{x^2}{2} \arccos(x) + C$$

Substituindo as constantes e os limites temos:

$$\text{Parte 2} = \frac{-16r^4}{R^2} \left[\frac{\pi}{24} \right] - 0 \left(\frac{1/2}{4\sqrt{1-1/4}} - 0 \right) + \frac{1}{8} \cdot \frac{\pi}{3} - 0$$

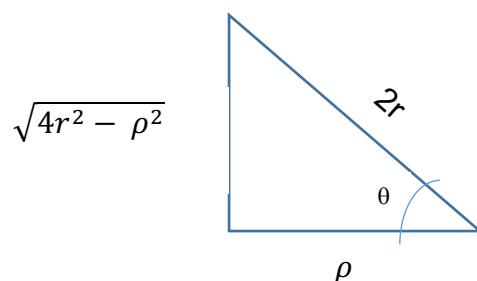
$$\text{Parte 2} = \frac{2r^4\pi}{3R^2} + \frac{16r^4}{R^2} \cdot \frac{\sqrt{3}}{12} - \frac{2\pi r^4}{3R^2}$$

$$\text{Parte 2} = \frac{4\sqrt{3}}{3} \frac{r^4}{R^2} - \frac{4r^4\pi}{3R^2}$$

$$\text{Parte 2} = \frac{4}{3} \frac{r^4}{R^2} (\sqrt{3} - \pi)$$

A.3 – Parte 3

$$\text{Parte 3} = \frac{r^2}{R^2} \int_0^r 2 \rho \operatorname{sen} \left(2 \arccos \frac{\rho}{2r} \right) d\rho$$



$$\cos \theta = \frac{\rho}{2r}$$

$$\operatorname{sen} \theta = \sqrt{1 - \left(\frac{\rho}{2r}\right)^2}$$

Então temos que:

$$\text{Parte 3} = \frac{r^2}{R^2} \int \frac{4 \rho^2}{2r} \sqrt{1 - \left(\frac{\rho}{2r}\right)^2} d\rho = \frac{2r}{R^2} \int \rho^2 \left(\sqrt{1 - \left(\frac{\rho}{2r}\right)^2} \right) d\rho$$

Sendo:

$$\operatorname{sen} x = \frac{\rho}{2r}$$

Para $\rho \rightarrow 0$ então $x \rightarrow 0$

Para $\rho \rightarrow r$ então $x \rightarrow \pi/6$

$$\operatorname{sen} 2x = 2 \operatorname{sen}(x) \cdot \cos(x)$$

$$\operatorname{sen} \left(2 \arccos \frac{\rho}{2r} \right) = 2 \sqrt{1 - \left(\frac{\rho}{2r} \right)^4} \frac{\rho}{2r}$$

$$\partial \rho = 2r \cos(x) \partial x$$

Então:

$$\text{Parte 3} = \frac{2r}{R^2} \int_0^{\pi/6} 4r^2 \operatorname{sen}(x) \cos(x) 2r \cos(x) \cdot \partial x$$

$$\text{Parte 3} = \frac{16 r^4}{R^2} \int_0^{\pi/6} \operatorname{sen}(x) \cdot \cos^2(x) \cdot \partial x$$

Fazendo

$$u = \cos(x) \quad \partial u = -\operatorname{sen}(x) \partial x$$

Para $x \rightarrow 0$ então $u \rightarrow 1$

Para $x \rightarrow \pi/6$ então $u \rightarrow \frac{\sqrt{3}}{2}$

Então:

$$\int \operatorname{sen}(x) \cdot \cos^2(x) \cdot \partial x = - \int u^2 \cdot \partial u = \frac{u^3}{3}$$

Portanto:

$$\text{Parte 3} = \frac{16 r^4}{R^2} \int_1^{\sqrt{3}/2} -u^2 \cdot \partial u = \frac{16 r^4}{R^2} \cdot \frac{u^3}{3} \Big|_{\sqrt{3}/2}^1$$

$$Parte\ 3 = \frac{16 r^4}{R^2} \left[\frac{1}{3} - \frac{3\sqrt{3}}{24} \right]$$

$$Parte\ 3 = 16 \cdot \frac{r^4}{R^2} \left[\frac{8 - 3\sqrt{3}}{24} \right]$$

$$Parte\ 3 = \frac{r^4}{R^2} \left[\frac{128 - 48\sqrt{3}}{24} \right]$$

$$Parte\ 3 = \frac{r^4}{R^2} \left[\frac{16}{3} - 2\sqrt{3} \right]$$

Desta forma a probabilidade de um nó sensor X_2 se comunicar com o nó sensor sink através do nó sensor X_1 , sendo que este nó sensor X_1 se comunica diretamente com S ($X_2 \rightarrow X_1 \rightarrow S$) em função de r e R é dada pela soma das três partes calculadas separadamente.

$$P \{X_2 \rightarrow X_1 \rightarrow S\} = Parte\ 1 + Parte\ 2 + Parte\ 3$$

$$P \{X_2 \rightarrow X_1 \rightarrow S\} = \frac{1}{\pi R^2} \left(\frac{\pi r^4}{R^2} + \frac{4 r^4}{3R^2} (\sqrt{3} - \pi) + \frac{r^4}{R^2} \left[\frac{16}{3} - 2\sqrt{3} \right] \right)$$

APÊNDICE B – SCRIPTS MATLAB

Este apêndice contém o código fonte do script descrito no capítulo 5 – seção 5.2.1- Script de Simulação e utilizado nas simulações descritas no capítulo 6 para se obter os vetores de perda de comunicação.

```

clear          %clear variables & items
clear all
clear global  %clear global variables
close all     %close windows
clc          %clear command window

%*****%
%   Variáveis do Modelo           %
%*****%

raio_area=      % Raio da Área de sensoriamento em metros
raio_alcance=   % raio de alcance do sensor.
relacao=
qtd_sensores=  % quantidade total de nós sensores excluindo o nó sink

A= % fator percentual de nós sensores que estão ativos.
    % Ex. 0,2 significa que 20% dos nós sensores definidos estão ativos.

sensores_ativos=A*qtd_sensores

qtd_execucoes = % quantidade vezes que a simulação será executada

```

```

%*****%
%          Variáveis utilizadas para cálculo          %
%
%          %
%          xl e x = coordenada x de cada nó sensor          %
%          yl e x = coordenada y de cada nó sensor          %
%          %
%          HR = hipotenusa do triângulo formado por cada nó. Usada no teste para          %
%          saber se o nó está dentro da área de Sensoriamento          %
%*****%

e=1;
MR=[];

while e <= qtd_execucoes

%*****%
%          Sorteio das coordenadas x e y de cada nó sensor.          %
%          a coordenada x deve estar dentro do range -R até R          %
%          Teste para validar se o sensor está dentro da Área de Sensoriamento          %
%*****%
    n=0;          %inicia o contador de nós
    i=1;          %inicia o índice de cada nó

    while n < sensores_ativos %executa este laço até ter N nós dentro de Asr

        xl(i)=(raio_area-(-1*raio_area)).*rand(1)+(-1*raio_area);
        coordenada_XS0=xl';

        yl(i)=(raio_area-(-1*raio_area)).*rand(1)+(-1*raio_area);
        coordenada_YS0=yl';

        hipotenusa_raio_area=sqrt((coordenada_XS0.^2)+(coordenada_YS0.^
2));

```

```

        if hipotenusa_raio_area <=raio_area.
            n=n+1;
            i=i+1;
        end
    end
clear n;
clear i;

%*****%
%   Teste de comunicação direta com sink   %
%*****%

p=1; % controla o conjunto s0
q=1; % controla o conjunto S0
r=1; % controla o conjunto S1

coordenada_xs0 = []; % reinicia a contagem do conjunto_s0
coordenada_XS1 = []; % reinicia a contagem do conjunto_S1

while q <= length (coordenada_XS0)
    hipotenusa_raio_sink =
sqrt((coordenada_XS0(q).^2)+(coordenada_YS0(q).^2));
    if hipotenusa_raio_sink < raio_alcance
        coordenada_xs0(p) = coordenada_XS0(q);
        coordenada_ys0(p) = coordenada_YS0(q);
        p=p+1;
    else
        coordenada_XS1(r) = coordenada_XS0(q);
        coordenada_YS1(r) = coordenada_YS0(q);
        r=r+1;
    end
    q=q+1;
end
c_s0(e) = length(coordenada_xs0);

```

```

c_S1(e) = length(coordenada_XS1);
conjunto_s0(e) = c_s0(e)';
conjunto_S1(e) = c_S1(e)';

%*****%
%   Teste de comunicação com mais saltos   %
%*****%

count=0;
VR=[length(coordenada_XS1),length(coordenada_xs0)];

if (c_s0(e) ~=0)
    coordenada_XS2=0;
    coordenada_YS2=0;
    coordenada_xs1=0;
    while (~isempty(coordenada_XS2)) && (~isempty(coordenada_xs1));
        [coordenada_XS2,coordenada_YS2,coordenada_xs1,coordenad
a_ys1] = conjunto
(coordenada_XS1,coordenada_YS1,coordenada_xs0,coordenada_ys0)
        ;
        coordenada_xs0 = coordenada_xs1;
        coordenada_ys0 = coordenada_ys1;
        coordenada_XS1 = coordenada_XS2;
        coordenada_YS1 = coordenada_YS2;

        VR=[VR,length(coordenada_XS2),length(coordenada_xs1)];
        count=count+1;
        length(coordenada_xs0);
        length(coordenada_XS1);
    end
end
end

```



```
MR{e}=VR;
e=e+1
end
```

```
maxLength = max(cellfun(@(x) numel(x), MR));
MR_auxiliar = cellfun(@(x) cat(2,x,zeros(1,maxLength-length(x))), MR, 'UniformOutput', false);
MR_matriz=cell2mat(MR_auxiliar);
xlswrite(filename, MR_matriz);
```

```
%*****%
% Função Conjunto %
%*****%
```

```
function [coordenada_XS2, coordenada_YS2, coordenada_xs1, coordenada_ys1] =
conjunto (coordenada_XS1, coordenada_YS1, coordenada_xs0, coordenada_ys0);
j=1;
k=1;
u=1;
v=1;
raio_alcance = 100;
coordenada_xs1=[];
coordenada_ys1=[];
coordenada_XS2=[];
coordenada_YS2=[];
    for j=1:length(coordenada_XS1);
        k=1;
        flag=0;
        while (k <= length(coordenada_xs0) & ~flag == 1)
```

```
        hipotenusa          =          sqrt(((coordenada_XS1(j)-
coordenada_xs0(k))^2)+((coordenada_YS1(j)-coordenada_ys0(k))^2));
        if hipotenusa < raio_alcance;
            flag=1;
            coordenada_xs1(v) = coordenada_XS1(j);
            coordenada_ys1(v) = coordenada_YS1(j);
            v=v+1;
        end
        k=k+1;

    end
    if flag==0
        coordenada_XS2(u) = coordenada_XS1(j);
        coordenada_YS2(u) = coordenada_YS1(j);
        u=u+1;
    end
end
```

APÊNDICE C – IMPLEMENTAÇÃO SIMULINK®

A Figura 50 apresenta o circuito implementado em *Simulink*® utilizado para a realização deste trabalho.

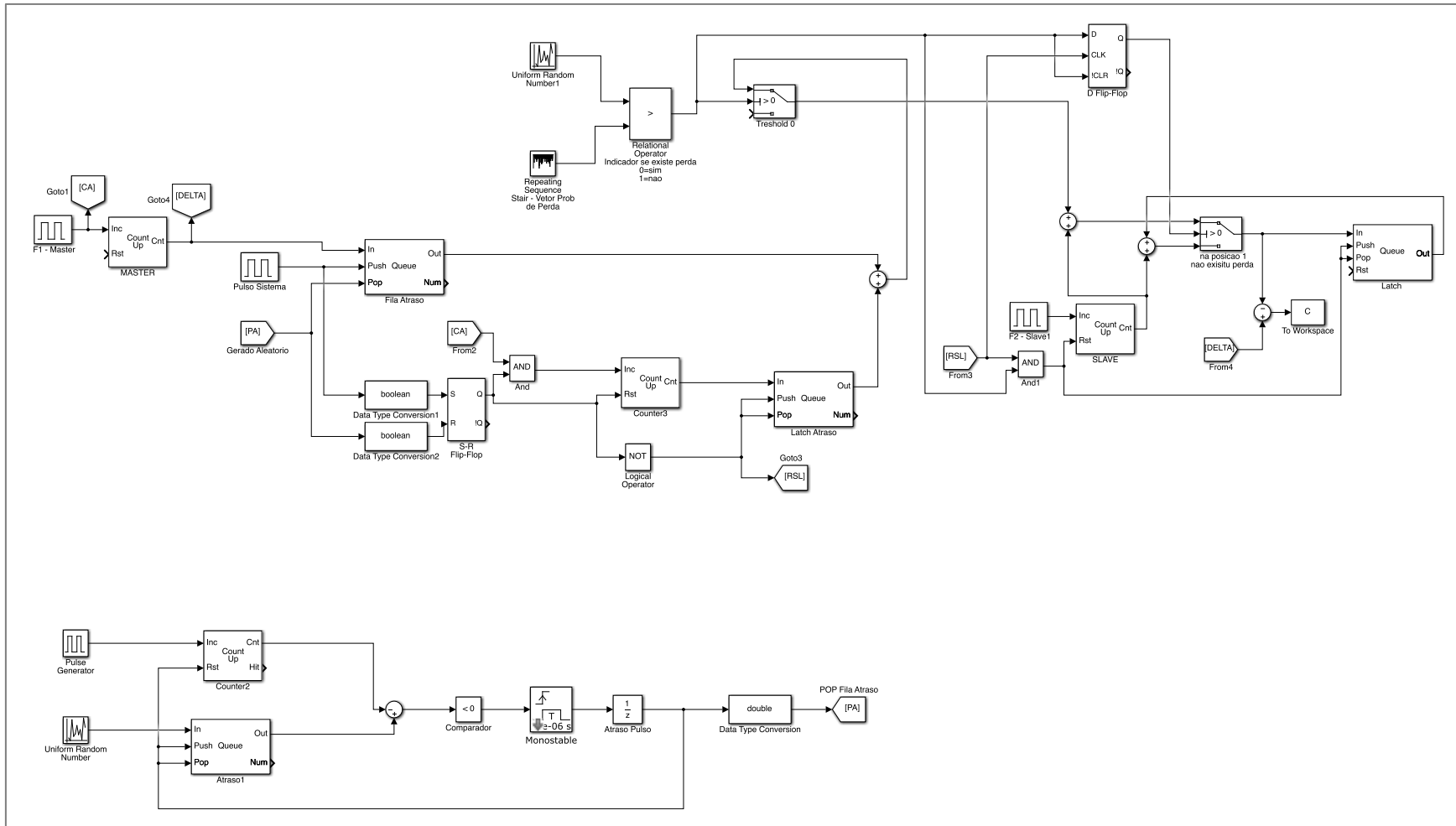


Figura 50 – Simulador
 Fonte: Elaborada pelo Autor

APÊNDICE D – VETORES DE PERDA

A Tabela 11 e a Tabela 12 apresentam os vetores de perda resultantes da execução do script descrito na seção 5.2.1- Script de Simulação e utilizados nas simulações descritas no capítulo 5.

Tabela 11 - Vetor de Perda - Intervalo de 4000 a 7300 nós

Número de Execuções	4000 nós	5000 nós	5500 nós	6000 nós	6500 nós	7000 nós	7300 nós
	Perda	Perda	Perda	Perda	Perda	Perda	Perda
1	0.98275	0.97380	0.56273	0.25333	0.10877	0.04186	0.04027
2	0.99225	0.89600	0.61055	0.18767	0.11785	0.02829	0.03589
3	0.99650	0.80760	0.41564	0.12550	0.06477	0.05457	0.03986
4	0.99125	1.00000	0.65091	0.97667	0.09523	0.02757	0.02288
5	0.99200	0.93520	0.73836	0.99900	0.12338	0.03943	0.03356
6	0.99050	0.75440	0.69036	0.99483	0.07938	0.03471	0.03000
7	0.99650	0.99780	0.78782	0.14900	0.09200	0.08286	0.02027
8	0.99025	0.88640	0.48164	0.33417	0.10123	0.04686	0.03384
9	0.98175	0.92940	0.99109	0.28017	0.29369	0.04143	0.02096
10	0.99000	0.99860	0.85273	0.34200	0.09954	0.03400	0.03151
11	0.99125	0.84940	0.64727	0.30383	0.99615	0.04414	0.02918
12	1.00000	0.99360	0.69655	0.25850	0.12923	0.03700	0.02959
13	0.99950	0.97720	0.98400	0.15833	0.10062	0.02857	0.03781
14	0.99075	0.83160	0.94600	0.10650	0.08277	0.06543	0.02658
15	0.99725	0.99700	0.90636	0.26700	0.09877	0.03529	0.01712
16	0.98650	0.94460	0.99818	0.09400	0.14077	0.07686	0.03055
17	0.99475	0.99920	0.77145	0.31983	0.22477	0.03600	0.04247
18	0.98575	0.93180	0.73382	0.20033	0.10354	0.06786	0.02589
19	0.99875	0.99320	0.69327	0.98517	0.99585	0.07514	0.02575
20	0.98200	0.98940	0.97964	0.39733	0.07569	0.03886	0.03151
21	0.99900	0.98300	0.72182	0.33867	0.07769	0.04071	0.01863
22	0.99925	0.99940	0.99745	0.43683	0.11692	0.03871	0.03466
23	0.99575	0.92640	0.99127	0.21283	0.06046	0.03700	0.04726
24	0.99300	0.99560	0.71600	0.33950	0.13615	0.08243	0.04014
25	0.98250	0.94080	0.99982	0.99883	0.99954	0.06843	0.02425
26	0.99975	0.90760	0.75036	0.21233	0.08138	0.04829	0.03918
27	0.99775	0.99720	0.99945	0.45367	0.08154	0.07371	0.01575
28	0.99925	0.89180	0.99782	0.27317	0.13015	0.05171	0.02411
29	0.99275	0.98760	0.96073	0.39967	0.12138	0.03600	0.03329
30	0.99975	0.86560	0.62655	0.23333	0.21431	0.06843	0.03918
31	0.99125	0.95860	0.95418	0.28000	0.09015	0.04000	0.02822
32	0.98775	0.96560	0.93873	0.13867	0.08215	0.08686	0.04836
33	0.99025	0.97740	0.50491	0.98267	0.07477	0.04129	0.02808
34	0.99500	0.96980	0.85673	0.09617	0.06969	0.05257	0.04411

35	0.99400	0.96380	0.96145	0.28533	0.06954	0.99971	0.03671
36	0.99925	0.97620	0.80818	0.51217	0.04892	0.04171	0.04836
37	0.99625	0.99980	0.86909	0.39083	0.16862	0.04114	0.05534
38	0.99850	0.96540	0.73982	0.56167	0.07508	0.03086	0.02589
39	1.00000	0.99840	0.97655	0.99983	0.06077	0.04343	0.02356
40	0.98950	0.96400	0.40327	0.52983	0.08215	0.02800	0.03877
41	0.99825	0.94140	0.61745	0.68317	0.08215	0.05229	0.02411
42	0.99775	0.85540	0.98673	0.13167	0.07431	0.05129	0.02247
43	0.97175	0.81060	0.97945	0.21433	0.08738	0.06143	0.03137
44	0.97800	0.99620	0.70836	0.38800	0.13569	0.04200	0.02000
45	0.99950	0.97160	0.99836	0.18050	0.09492	0.04429	0.04589
46	0.99400	0.99980	0.82618	0.18333	0.06369	0.07557	0.03863
47	0.99500	0.99800	0.99964	0.16533	0.05631	0.06614	0.02342
48	0.99475	0.99880	0.91382	0.74233	0.08000	0.03414	0.02712
49	0.96850	0.99120	0.70909	0.22550	0.06215	0.05414	0.01849
50	1.00000	0.98800	0.65018	1.00000	0.09431	0.03857	0.03932
51	0.99675	0.98060	0.99400	0.98917	0.11031	0.03471	0.03534
52	0.97200	0.91740	0.57109	0.36567	0.05000	0.10029	0.03205
53	0.99350	1.00000	0.83018	0.43017	0.05862	0.04329	0.02630
54	0.98200	0.96980	0.99873	0.37350	0.06677	0.05343	0.01918
55	0.97225	0.90160	0.99345	0.48017	0.10585	0.02929	0.02178
56	0.99375	0.96700	0.94673	0.19183	0.22400	0.03929	0.02123
57	0.98175	0.78180	0.55636	0.16550	0.30554	0.03443	0.01795
58	0.99925	1.00000	0.99164	0.27750	0.09508	0.04129	0.01301
59	0.99500	0.98000	0.91873	0.48950	0.09538	0.05114	0.03438
60	0.98825	0.78120	0.93273	0.57800	0.10846	0.05386	0.04548
61	0.99250	0.93360	0.86618	0.21283	0.11000	0.05100	0.02822
62	0.98600	0.87680	0.66945	0.13433	0.15538	0.06214	0.03315
63	0.98725	0.97920	0.94927	0.38233	0.08600	0.06243	0.02548
64	0.99800	0.87700	0.68964	0.48933	0.07092	0.06971	0.02425
65	0.98625	0.99940	0.87436	0.32600	0.08600	0.03514	0.03329
66	0.96500	0.99460	0.70855	0.96783	0.06908	0.03943	0.02384
67	0.99675	0.93620	0.99018	0.22833	0.11938	0.06314	0.04507
68	0.98925	0.93700	0.93945	0.35467	0.10092	0.04529	0.06644
69	0.99775	0.98280	0.72473	0.24183	0.06723	0.03900	0.02589
70	0.99900	0.99280	0.76582	0.22700	0.06631	0.02929	0.03712
71	0.99650	0.82600	0.99909	0.31700	0.10231	0.03171	0.04712
72	0.99450	0.97600	0.73255	0.99850	0.07462	0.09843	0.04945
73	0.98975	0.97540	0.62945	0.99500	0.06508	0.03057	0.02726
74	0.99875	0.97860	0.94091	0.30800	0.09569	0.03614	0.02425
75	0.99375	0.94640	0.71400	0.38417	0.08569	0.05371	0.02658
76	0.99700	0.94780	0.74364	0.23700	0.07800	0.05286	0.03137
77	0.99700	0.99440	0.99873	0.61733	0.10585	0.03414	0.04068
78	1.00000	0.97700	0.70382	0.15617	0.10615	0.02786	0.03178
79	0.99500	0.98080	0.93345	0.19333	0.65646	0.04657	0.03329
80	0.98900	0.97900	0.49982	0.25300	0.20477	0.03557	0.02301
81	0.98650	0.95980	0.79400	0.15000	0.08862	0.05329	0.04699
82	0.99325	0.98560	0.96400	0.49867	0.08954	0.04071	0.03959
83	0.98125	0.98360	0.99200	0.14800	0.06769	0.99986	0.03000
84	0.96775	1.00000	0.97982	0.30300	0.10415	0.07814	0.02205
85	0.98875	0.99580	0.96091	0.31867	0.10508	0.05686	0.03479

86	0.97975	0.96780	0.55782	0.33417	0.07492	0.02729	0.03795
87	0.99700	0.89180	0.97236	0.34567	0.13446	0.06100	0.01740
88	1.00000	0.99960	0.66673	0.20650	0.05323	0.04157	0.03014
89	0.97800	0.98980	0.81600	0.48633	0.10415	0.04071	0.03466
90	0.98550	0.99420	0.76109	0.28583	0.08754	0.03357	0.03849
91	0.99550	0.94360	0.77327	0.41233	0.06492	0.06071	0.03068
92	0.97975	0.92180	0.91109	0.36817	0.14092	0.99443	0.02849
93	0.99275	0.99940	0.67400	0.18917	0.13262	0.03829	0.02342
94	0.99850	0.98480	0.93909	0.94067	0.06877	0.04000	0.02808
95	0.98125	0.99980	0.99273	0.17833	0.08923	0.03671	0.02890
96	0.99975	0.90040	0.64873	0.24917	0.10046	0.03614	0.01808
97	0.99450	0.99760	0.93400	0.99850	0.19785	0.03914	0.03479
98	0.97850	0.99940	0.65564	0.99817	0.10538	0.03257	0.02753
99	0.97400	0.87440	0.67873	1.00000	0.99569	0.02714	0.02603
100	0.98025	0.90680	0.85127	0.20300	0.10523	0.04186	0.03288
101	0.99350	0.98540	0.99364	0.23500	0.07985	0.04571	0.04945
102	0.99875	0.98080	0.58036	0.11517	0.14415	0.04414	0.02575
103	0.99475	0.99800	0.65545	0.48767	0.08692	0.05600	0.02795
104	0.99275	0.98780	0.79345	0.17800	0.06031	0.05100	0.04356
105	0.98250	0.99480	0.71127	0.27417	0.99969	0.06071	0.02808
106	0.99750	1.00000	0.99727	0.48583	0.09415	0.03286	0.03233
107	0.99625	0.99020	0.96818	0.21133	0.11415	0.04943	0.02548
108	0.95125	0.86240	0.84745	0.38450	0.07554	0.04771	0.02616
109	0.99575	0.99000	0.75109	0.35650	0.99908	0.04343	0.03192
110	0.98775	0.99880	0.99145	0.40900	0.07508	0.03586	0.02973
111	0.99500	0.99260	0.70255	0.29117	0.08985	0.04886	0.02712
112	0.99875	0.99920	0.64491	0.23083	0.10015	1.00000	0.03014
113	0.98425	0.85080	0.76982	0.29483	0.08462	0.04957	0.03370
114	0.97750	0.99660	0.91527	0.48350	0.05415	0.05786	0.99986
115	0.99900	0.96840	0.64655	0.99133	0.16492	0.04571	0.02452
116	0.99950	0.88260	0.99764	0.49383	0.07677	0.04414	0.03233
117	0.99950	0.86160	0.45655	0.18717	0.10154	0.04529	0.03603
118	0.99925	0.89000	0.98673	0.20767	0.12400	0.05929	0.05096
119	0.99325	0.87140	0.80127	0.99833	0.13323	0.04357	0.02904
120	0.99900	0.96720	0.92545	0.37533	0.07662	0.03543	0.04452
121	0.99875	0.99700	0.83436	0.65017	0.17108	0.03729	0.02247
122	0.99150	0.98140	0.62782	0.29650	0.05923	0.04800	0.02795
123	0.97850	1.00000	0.63400	0.47967	0.14631	0.07657	0.02986
124	0.98650	0.99760	0.69291	0.44800	0.09123	0.06529	0.03164
125	0.99850	0.92080	0.43818	0.22350	0.08415	0.99886	0.04178
126	0.99025	0.99500	0.83345	0.35383	0.10446	0.04729	0.02685
127	0.99975	0.99940	0.84545	0.28100	0.05277	0.04586	0.02740
128	0.99900	0.99620	0.88891	0.98833	0.12462	0.03757	0.03849
129	0.99400	0.88080	0.99436	0.43433	0.32508	0.07229	0.02562
130	0.99325	0.87500	0.89636	0.99917	0.15015	0.04200	0.01781
131	0.99500	0.97200	0.63273	0.14150	0.11600	0.03914	0.02945
132	0.99900	0.99300	0.99964	0.19833	0.12477	0.03843	0.02397
133	0.99350	0.98060	0.84164	0.34600	0.11615	0.05171	0.03575
134	0.99900	0.96400	0.66018	0.20233	0.14446	0.05229	0.03247
135	0.99825	0.98820	0.40309	0.25117	0.06631	0.04286	0.03411
136	0.94275	0.97260	0.98691	0.31583	0.17015	0.04157	0.01712

137	0.99500	0.99920	0.75982	0.24250	0.31338	0.04357	0.01685
138	0.95875	0.96700	0.84109	0.19817	0.09892	0.02943	0.02589
139	0.97600	0.99960	0.63782	0.99900	0.99954	0.05643	0.02548
140	0.99775	0.91200	0.99418	0.36833	0.20723	0.03514	0.05014
141	0.99325	0.99620	0.55800	0.58900	0.09631	0.05043	0.01699
142	0.99725	0.88140	0.69618	0.15383	0.10892	0.05471	0.02178
143	0.99775	0.89940	0.90182	0.79083	0.11554	0.08343	0.03096
144	0.99900	0.98740	0.35182	0.13367	0.09338	0.03514	0.02630
145	0.99975	0.96740	0.99982	0.18267	0.14508	0.05143	0.03233
146	0.98675	0.98220	0.99018	0.99967	0.05646	0.06814	0.02575
147	0.99625	0.96320	0.94400	0.19100	0.10292	0.04214	0.02836
148	0.99125	0.99840	0.53145	0.27883	0.06062	0.03129	0.01918
149	0.99950	0.99680	0.99182	0.49517	0.16215	0.06743	0.03493
150	0.99975	0.75460	0.99727	0.12133	0.12446	0.02743	0.04260
151	0.99975	0.92780	0.63818	0.20567	0.07969	0.02857	0.03534
152	0.99725	0.89720	0.99964	0.28033	0.10185	0.06543	0.03288
153	0.99900	0.89720	0.99764	0.45083	0.05723	0.03600	0.04726
154	0.99750	0.90240	0.94909	0.14717	0.15015	0.04600	0.02945
155	0.95900	0.95460	0.53091	0.26733	0.18415	0.04514	0.05438
156	0.97575	0.99620	0.54836	0.19883	0.07415	0.05086	0.02712
157	0.99050	0.95720	0.52945	0.20683	0.10154	0.03243	0.03178
158	0.97000	0.99440	1.00000	0.34833	0.10277	0.03086	0.03082
159	0.99975	0.98420	0.35236	0.62550	0.10185	0.02471	0.02795
160	0.99950	0.89760	0.46436	0.50000	0.06323	0.06514	0.03110
161	1.00000	0.89240	0.70909	0.99900	0.07338	0.06071	0.01959
162	0.98450	0.93220	0.71418	0.69250	0.05585	0.02757	0.02877
163	0.97200	0.99920	0.95036	0.99133	0.11523	0.04443	0.02890
164	0.97200	0.78880	0.96691	0.29917	0.13815	0.05071	0.04000
165	0.97875	0.94500	0.99127	0.33183	0.99954	0.04114	0.03205
166	0.99550	0.96300	0.75000	0.60450	0.06062	0.02929	0.02753
167	0.99900	0.90420	0.73873	0.99983	0.10892	0.03943	0.03151
168	0.99250	0.91080	0.71473	0.99883	0.07846	0.04429	0.04137
169	0.99700	0.86000	0.43091	0.23433	0.10354	0.02643	0.04096
170	0.97675	0.98180	0.78200	0.27067	0.07323	0.04571	0.03041
171	0.99250	0.98500	0.72545	0.74850	0.13338	0.05529	0.99877
172	0.97250	0.96680	0.95673	0.99733	0.10908	0.06443	0.06438
173	0.97075	0.96940	0.97273	0.62683	0.99769	0.05114	0.03575
174	0.99650	0.95960	0.99709	0.67367	0.05323	0.05386	0.99959
175	0.98675	0.94520	0.44182	0.41600	0.12185	0.03243	0.02945
176	0.98450	0.98840	0.65964	0.22250	0.09062	0.02857	0.03849
177	1.00000	0.98720	0.63127	0.19533	0.11769	0.04043	0.04945
178	0.95325	0.93520	0.93000	0.38950	0.04785	0.03114	0.02068
179	0.99875	0.80320	0.58800	0.29933	0.06077	0.04943	0.03452
180	0.99425	0.96220	0.97945	0.19317	0.07415	0.06400	0.02616
181	0.97900	0.98320	0.78764	0.20033	0.08938	0.03814	0.03000
182	0.96450	0.90020	0.99218	0.19433	0.09723	0.05114	0.01658
183	0.98900	0.86120	0.81291	0.60150	0.98862	0.02414	0.03479
184	0.99575	0.96060	0.73691	0.14150	0.09846	0.03843	0.03411
185	0.99625	0.99800	0.53127	0.27150	0.06246	0.07457	0.02904
186	0.99300	0.97660	0.99891	0.13283	0.08600	0.04100	0.02753
187	0.99950	0.94080	0.98200	0.18133	0.05954	0.03700	0.04945

188	0.96525	0.88020	0.99727	0.13750	0.10415	0.05200	0.02123
189	0.97625	0.99680	0.99655	0.48217	0.08538	0.05343	0.02753
190	0.99500	0.99720	0.74618	0.25617	0.11277	0.03071	0.03096
191	0.98400	0.97380	0.93818	0.42233	0.08738	0.04371	0.02219
192	0.99675	0.86180	0.63909	0.61800	0.08046	0.03229	0.03247
193	0.99900	0.97800	0.99691	0.37017	0.10646	0.03200	0.01370
194	0.99350	0.99820	0.80291	0.65667	0.12508	0.04143	0.02945
195	1.00000	0.99960	0.64527	0.99800	0.07954	0.03543	0.02781
196	0.99950	0.99940	0.49927	0.18017	0.08154	0.05229	0.05192
197	0.99700	0.82740	0.82491	0.51867	0.09246	0.04471	0.02959
198	0.99400	0.92260	0.60473	0.74950	0.08554	0.03971	0.03425
199	0.99700	0.81700	0.82782	0.11683	0.08046	0.05757	0.02041
200	0.99850	0.70720	0.71982	0.33867	0.11754	0.03729	0.01973
Média Taxa de Perda	4000 nós	5000 nós	5500 nós	6000 nós	6500 nós	7000 nós	7300 nós
	99.04%	94.85%	79.95%	41.67%	15.12%	7.04%	4.59%

Fonte: Elaborada pelo Autor

Tabela 12 - Vetor de Perda - Intervalo de 7500 a 10000 nós

Número de Execuções	7500 nós	7700 nós	8000 nós	8500 nós	9000 nós	10000 nós
	Perda	Perda	Perda	Perda	Perda	Perda
1	0.02413	0.01519	0.01288	0.00824	0.00856	0.00470
2	0.01853	0.01403	0.01363	0.00918	0.00389	0.00300
3	0.02120	0.03065	0.00988	0.00376	0.00378	0.00110
4	0.01973	0.02221	0.00838	0.00729	0.00456	0.00170
5	0.02400	0.02494	0.01475	0.00988	0.00289	0.00080
6	0.04173	0.02052	0.01875	0.00929	0.00533	0.00220
7	0.99907	0.01935	0.01863	0.00718	0.00456	0.00150
8	0.01933	0.01870	0.01125	0.00941	0.00400	0.00220
9	0.01947	0.02597	0.00963	0.00400	0.00344	0.00560
10	0.02347	0.01234	0.01938	0.00953	0.00489	0.00130
11	0.02947	0.02130	0.00800	0.00729	0.00433	0.00080
12	0.01413	0.01948	0.01813	0.01047	0.00689	0.00230
13	0.01427	0.02221	0.02725	0.00588	0.00333	0.00390
14	0.02400	0.03636	0.01800	0.00906	0.00511	0.00710
15	0.02133	0.02481	0.01125	0.00906	0.00411	0.00230
16	0.02173	0.01390	0.01488	0.00729	0.00522	0.00380
17	0.01773	0.02065	0.01263	0.00635	0.00433	0.00310
18	0.01827	0.02273	0.01163	0.00682	0.00611	0.00100
19	0.03053	0.02597	0.01550	0.00718	0.00622	0.00330
20	0.02307	0.01416	0.02250	0.00729	0.00433	0.00230
21	0.01893	0.01610	0.02813	0.00741	0.00667	0.00220
22	0.02187	0.99974	0.02213	0.00565	0.00422	0.00170
23	0.02627	0.01857	0.01688	0.00906	0.00522	0.00270
24	0.03053	0.02844	0.01075	0.00718	0.00489	0.00160
25	0.03053	0.03039	0.01225	0.00694	0.00767	0.00250

26	0.03467	0.02494	0.01050	0.01094	0.00544	0.00060
27	1.00000	0.01909	0.01925	0.00776	0.00356	0.00130
28	0.02027	0.01636	0.01300	0.00388	0.00333	0.00290
29	0.01627	0.01468	0.01200	0.00694	0.01122	0.00320
30	0.02307	0.02195	0.01063	0.00765	0.00767	0.00290
31	0.02173	0.01857	0.02438	0.00718	0.00233	0.00190
32	0.02960	0.01532	0.02150	0.01071	0.00433	0.00240
33	0.02187	0.02000	0.01250	0.00588	0.00778	0.00180
34	0.03280	0.99948	0.01613	0.00765	0.00667	0.00240
35	0.02053	0.02169	0.01350	0.00929	0.00267	0.00110
36	0.01507	0.99987	0.01488	0.00565	0.01189	0.00230
37	0.02427	0.01701	0.01063	0.01529	0.00567	0.00210
38	0.02960	0.02325	0.00600	0.00859	0.00667	0.00150
39	0.02533	0.02286	0.02000	0.01106	0.00711	0.00450
40	0.01933	0.03273	0.01550	0.00671	0.00511	0.00220
41	0.01800	0.01494	0.01525	0.00776	0.00444	0.00250
42	0.01867	0.01662	0.02775	0.00482	0.99978	0.00340
43	0.01960	0.01857	0.01725	0.00576	0.00444	0.00150
44	0.02107	0.02312	0.01725	0.00788	0.00400	0.00160
45	0.02320	0.02844	0.01538	0.00671	0.00778	0.00280
46	0.01773	0.01792	0.00963	0.00976	0.00589	0.00140
47	0.02027	0.01701	0.01413	0.00824	0.00556	0.00060
48	0.02227	0.01455	0.01425	0.00694	0.00556	0.00070
49	0.02093	0.01662	0.02363	0.00976	0.00333	0.00180
50	0.02867	0.01896	0.01450	0.00518	0.00489	0.00230
51	0.02933	0.02623	0.01275	0.00518	0.00756	0.00320
52	0.99880	0.02455	0.01000	0.00600	0.00589	0.00190
53	0.03133	0.01299	0.00575	0.00765	0.00333	0.00310
54	0.02840	0.01260	0.02075	0.01024	0.00589	0.00210
55	0.03133	0.01779	0.02075	0.00788	0.00589	0.00340
56	0.02253	0.01377	0.01663	0.01318	0.00400	0.00340
57	0.01600	0.02013	0.01638	0.00624	0.00544	0.00210
58	0.05040	0.02247	0.03075	0.00435	0.01022	0.00070
59	0.03160	0.01974	0.00950	0.00988	0.00600	0.00320
60	0.02453	0.01506	0.01250	0.01012	0.00644	0.00130
61	0.02227	0.01727	0.01513	0.00753	0.00233	0.00220
62	0.02200	0.01727	0.01925	0.00588	0.00400	0.00290
63	0.02013	0.02078	0.01075	0.00729	0.00489	0.00220
64	0.01760	0.01779	0.00913	0.00776	0.00689	0.00200
65	0.03227	0.01130	0.01463	0.00494	0.00544	0.00360
66	0.02747	0.01974	0.02025	0.00765	0.00889	0.00190
67	0.01573	0.02325	0.01563	0.00376	0.01056	0.00240
68	0.02880	0.01545	0.01125	0.01094	0.00478	0.00170
69	0.01733	0.01974	0.01875	0.00541	0.00444	0.00090
70	0.01440	0.02416	0.01875	0.01035	0.00256	0.00270
71	0.03400	0.02948	0.01388	0.00965	0.00156	0.00190
72	0.04480	0.01273	0.01113	0.00800	0.00344	0.00180
73	0.01973	0.02481	0.01363	0.00835	0.00389	0.00100
74	0.03627	0.02026	0.01200	0.00859	0.00478	0.00230
75	0.01867	0.01987	0.01238	0.00682	0.00433	0.00200
76	0.02773	0.99961	0.01863	0.00976	0.00589	0.00310

77	0.03467	0.03000	0.01013	0.00435	0.00611	0.00150
78	0.03040	0.01610	0.01713	0.00506	0.00622	0.00350
79	0.04440	0.01948	0.01213	0.01506	0.00389	0.00130
80	0.02187	0.01558	0.01863	0.00741	0.01033	0.00170
81	0.02413	0.02987	0.02013	0.00812	0.00811	0.00190
82	0.03480	0.02571	0.01150	0.00624	0.00411	0.00230
83	0.02427	0.01636	0.01188	0.00859	0.00589	0.00280
84	0.02667	0.01688	0.01325	0.00835	0.00444	0.00450
85	0.01867	0.01442	0.01400	0.00894	0.00656	0.00180
86	0.00960	0.01286	0.01100	0.00376	0.00222	0.00090
87	0.02947	0.01558	0.01350	0.00765	0.00389	0.00250
88	0.03200	0.02026	0.00938	0.01141	0.00578	0.00240
89	0.01773	0.01442	0.01463	0.00894	0.00444	0.00210
90	0.04067	0.03195	0.01450	0.00647	0.00756	0.00270
91	0.03547	0.01532	0.01088	0.00647	0.00444	0.00290
92	0.02360	0.01948	0.01950	0.00906	0.00522	0.00440
93	0.02053	0.02325	0.01200	0.00424	0.00589	0.00190
94	0.02560	0.02065	0.01138	0.00635	0.00467	0.00170
95	0.02627	0.01714	0.01125	0.01165	0.00511	0.00550
96	0.01587	0.01182	0.01175	0.00753	0.00344	0.00250
97	0.01880	0.00974	0.01950	0.00506	0.00678	0.00290
98	0.02133	0.01130	0.01325	0.00953	0.00667	0.00520
99	0.01933	0.01623	0.01375	0.00671	0.00589	0.00230
100	0.03800	0.01662	0.02263	0.00612	0.00711	0.00340
101	0.02347	0.00961	0.00375	0.01129	0.00622	0.00150
102	0.02173	0.01571	0.01388	0.01435	0.00467	0.00350
103	0.02413	0.01909	0.02250	0.00894	0.00456	0.00170
104	0.01400	0.02156	0.99988	0.00906	0.00456	0.00580
105	0.02120	0.01338	0.01688	0.00847	0.00411	0.00270
106	0.03120	0.02143	0.01225	0.01094	0.00478	0.00190
107	0.02280	0.00987	0.00675	0.00682	0.00722	0.00120
108	0.02333	0.03143	0.01225	0.00529	0.00289	0.00150
109	0.02320	0.01818	0.01325	0.01035	0.00411	0.00160
110	0.01733	0.01390	0.01275	0.01071	0.00344	0.00400
111	0.01587	0.02273	0.00950	0.99953	0.00611	0.00270
112	0.02627	0.02273	0.00938	0.00765	0.00500	0.00330
113	0.05427	0.01104	0.01625	0.02165	0.00567	0.00130
114	0.01400	0.02065	0.01950	0.02035	0.00389	0.00280
115	0.01920	0.02338	0.00688	0.00647	0.00622	0.00250
116	0.02160	0.01234	0.01238	0.00600	0.00433	0.00150
117	0.02067	0.00961	0.01213	0.00682	0.00444	0.00180
118	0.01733	0.01766	0.01500	0.01071	0.00189	0.00090
119	0.03173	0.01870	0.01775	0.00835	0.00467	0.00190
120	0.02787	0.02519	0.01188	0.00482	0.00633	0.00180
121	0.03733	0.02338	0.02013	0.00729	0.00467	0.00260
122	0.02987	0.01883	0.00638	0.00776	0.00489	0.00170
123	0.03600	0.01714	0.01488	0.00976	0.00556	0.00130
124	0.01693	0.01818	0.01425	0.00976	0.01011	0.00230
125	0.01827	0.02506	0.01325	0.00847	0.00167	0.00290
126	0.01787	0.01623	0.01200	0.00788	0.00344	0.00280
127	0.02067	0.02519	0.01150	0.00624	0.00744	0.00150

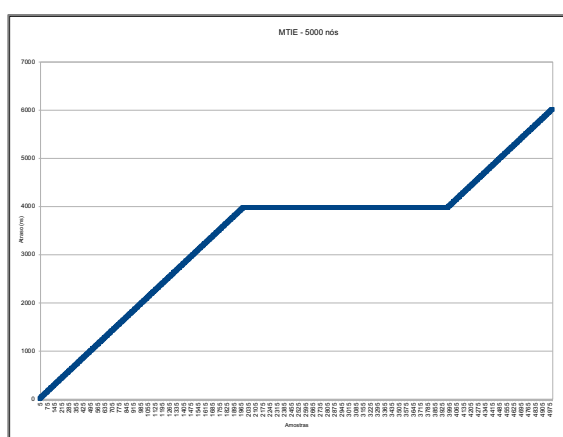
128	0.02267	0.02597	0.02275	0.01294	0.00667	0.00150
129	0.01973	0.01247	0.01650	0.01000	0.00944	0.00390
130	0.02547	0.01779	0.01263	0.00659	0.00433	0.00180
131	0.02027	0.01792	0.01413	0.00612	0.00500	0.00200
132	0.02720	0.02870	0.01425	0.00929	0.01322	0.00120
133	0.03467	0.03013	0.00975	0.00812	0.00411	0.00230
134	0.02213	0.01675	0.01725	0.00600	0.00333	0.00210
135	0.03107	0.02260	0.01000	0.00600	0.00422	0.00240
136	0.01800	0.02442	0.01388	0.00776	0.00078	0.00200
137	0.02000	0.01312	0.01688	0.00812	0.00667	0.00060
138	0.01747	0.01688	0.01638	0.00635	0.00511	0.00190
139	0.02520	0.01273	0.01975	0.01435	0.00378	0.00210
140	0.02013	0.01156	0.01425	0.00671	0.00356	0.00340
141	0.02000	0.03052	0.01463	0.00729	0.00500	0.00090
142	0.02333	0.02117	0.01838	0.01200	0.00467	0.00070
143	0.02240	0.01883	0.01200	0.00565	0.00300	0.00370
144	0.01720	0.01922	0.01075	0.00929	0.00411	0.00180
145	0.01973	0.01662	0.01288	0.00588	0.00478	0.00070
146	0.01693	0.02455	0.01175	0.00765	0.01089	0.00250
147	0.03333	0.01831	0.01088	0.00682	0.00422	0.00330
148	0.01413	0.01403	0.00938	0.00706	0.00322	0.00260
149	0.02320	0.01909	0.01213	0.00941	0.00511	0.00240
150	0.02027	0.02714	0.01163	0.00435	0.00556	0.00310
151	0.03480	0.02584	0.01438	0.00776	0.00456	0.00110
152	0.01800	0.01325	0.01050	0.00835	0.00644	0.00330
153	0.02320	0.01416	0.00988	0.00435	0.00444	0.00140
154	0.02813	0.02182	0.01563	0.01318	0.00256	0.00140
155	0.02533	0.02831	0.00975	0.00671	0.00422	0.00190
156	0.01827	0.01753	0.01125	0.00541	0.00278	0.00200
157	0.02173	0.01442	0.01688	0.00859	0.00467	0.00250
158	0.01307	0.02130	0.01075	0.00976	0.00700	0.00270
159	0.01627	0.01325	0.01450	0.00847	0.00600	0.00240
160	0.01867	0.01221	0.01188	0.00788	0.00544	0.00470
161	0.01840	0.01662	0.00988	0.00647	0.00289	0.00240
162	0.03253	0.03260	0.00538	0.00506	0.00622	0.00240
163	0.01733	0.02558	0.01313	0.01329	0.00622	0.00210
164	0.03600	0.02156	0.01538	0.00435	0.00844	0.00100
165	0.99960	0.01221	0.01000	0.00812	0.00778	0.00140
166	0.01587	0.01701	0.01363	0.00729	0.00422	0.00240
167	0.03347	0.00805	0.01013	0.01000	0.00411	0.00460
168	0.02493	0.02312	0.01738	0.01247	0.00367	0.00210
169	0.02960	0.02169	0.01350	0.00776	0.00589	0.00140
170	0.01813	0.03234	0.00463	0.00988	0.00433	0.00200
171	0.01520	0.01442	0.01425	0.00424	0.00611	0.00120
172	0.02627	0.01857	0.00838	0.00647	0.00456	0.00130
173	0.03373	0.01299	1.00000	0.00918	0.00211	0.00070
174	0.02493	0.01649	0.01275	0.00859	0.00467	0.00180
175	0.02693	0.02234	0.01275	0.00894	0.00500	0.00050
176	0.04373	0.02286	0.00825	0.00800	0.00456	0.00400
177	0.02280	0.03247	0.01188	0.00729	0.00600	0.00330
178	0.02640	0.02857	0.01575	0.00671	0.00911	0.00330

179	0.02267	0.01974	0.02450	0.00753	0.00433	0.00240
180	0.02933	0.02195	0.01275	0.01318	0.00578	0.00080
181	0.02187	0.01753	0.02263	0.00612	0.00444	0.00180
182	0.02613	0.01688	0.00750	0.00447	0.00622	0.00130
183	0.02973	0.01766	0.01713	0.01141	0.00322	0.00370
184	0.03000	0.01610	0.01175	0.00576	0.00544	0.00220
185	0.02360	0.01766	0.01125	0.00800	0.00533	0.00140
186	0.01560	0.01701	0.01613	0.00671	0.00389	0.00090
187	0.02067	0.02623	0.01888	0.00635	0.00511	0.00130
188	0.01773	0.03506	0.01613	0.00988	0.00611	0.00160
189	0.02733	0.01558	0.01500	0.01212	0.00633	0.00080
190	0.02373	0.02260	0.01275	0.00576	0.00367	0.00040
191	0.03507	0.01545	0.00750	0.00800	0.00622	0.00190
192	0.02747	0.01948	0.01575	0.00753	0.00511	0.00260
193	0.03000	0.01805	0.02025	0.01094	0.00533	0.00130
194	0.02080	0.01649	0.01938	0.01294	0.00444	0.00160
195	0.01733	0.01714	0.01813	0.00612	0.00900	0.00460
196	0.03680	0.02987	0.02113	0.00765	0.00478	0.00370
197	0.02027	0.01403	0.01738	0.00929	0.00611	0.00170
198	0.01680	0.00909	0.01175	0.00847	0.00322	0.00180
199	0.02467	0.01675	0.01400	0.00400	0.00300	0.00210
200	0.02027	0.01545	0.00963	0.00882	0.00633	0.00150
Média Taxa de Perda	7500 nós	7700 nós	8000 nós	8500 nós	9000 nós	10000 nós
	4.38%	3.92%	2.41%	1.31%	1.02%	0.23%

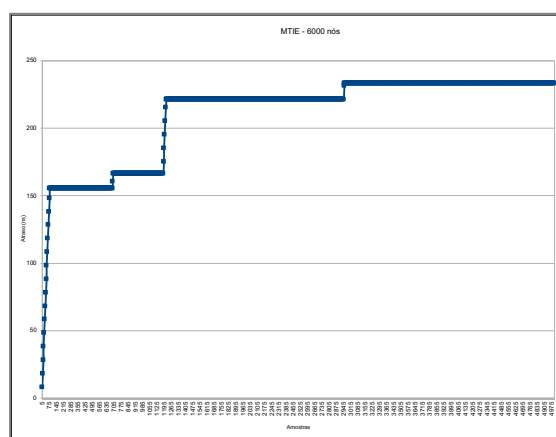
Fonte: Elaborada pelo Autor

APÊNDICE E – GRÁFICOS MTIE

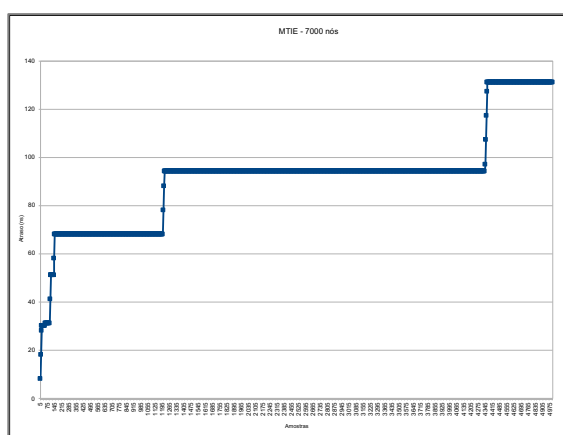
Neste apêndice a Figura 51 e Figura 52 apresentam os gráficos para o MTIE para as simulações contendo 5000, 6000, 7000, 7300, 7700, 8500 e 9000 nós sensores, conforme descrito na seção 6.3.1 Cenário 1 – Precisão do Sincronismo em Função da Quantidade de Nós Sensores, capítulo 6.



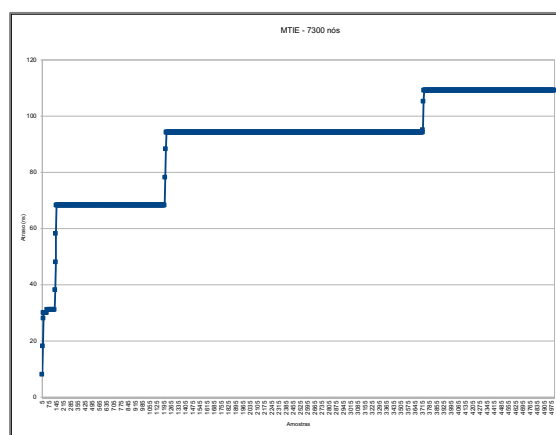
5000 sensores



6000 sensores



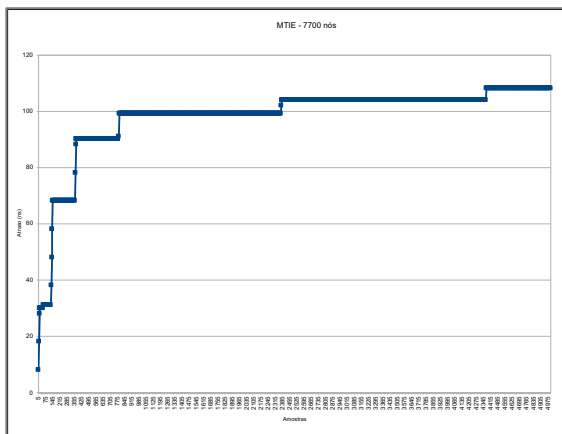
7000 sensores



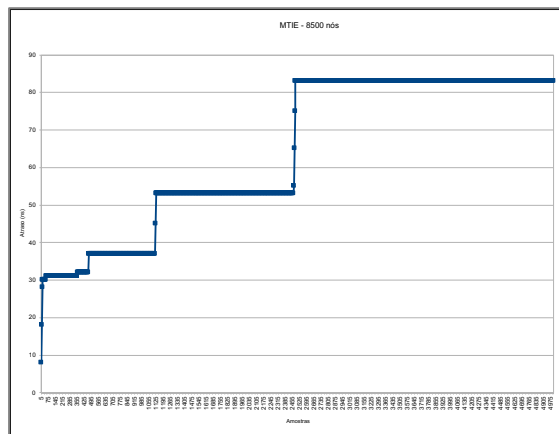
7300 sensores

Figura 51 - Gráfico MTIE - 5000 / 6000 / 7000 / 7300 sensores

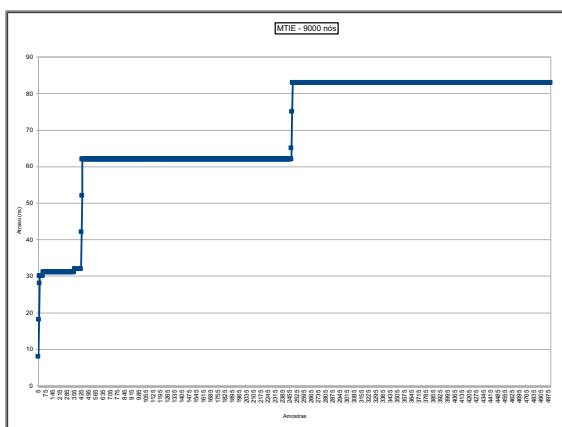
Fonte: Elaborada pelo Autor



7700 sensores



8500 sensores



9000 sensores

Figura 52 - Gráfico MTIE - 7700 / 8500 / 9000 sensores

Fonte: Elaborada pelo Autor