



DISSERTAÇÃO DE GRADUAÇÃO

**Uso do Simulador Cooja Contiki em
Redes de Sensores Sem Fio (RSSFs) para
Sistema de Telemetria em Hidrômetros**

Webert José Francisco Rodrigues Cunha

Brasília, março de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE GRADUAÇÃO

**Uso do Simulador Cooja Contiki em
Redes de Sensores Sem Fio (RSSFs) para
Sistema de Telemetria em Hidrômetros**

Webert José Francisco Rodrigues Cunha

*Dissertação de Graduação submetida ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Graduação em Engenharia Elétrica*

Banca Examinadora

Prof. Dr. José Camargo da Costa, FT/UnB
Orientador

Prof. PhD. Marcelo M. Carvalho
Examinador interno

Profa. Dr. Leticia Toledo Maia Zoby
Examinador interno

FICHA CATALOGRÁFICA

CUNHA, WEBERT JOSÉ FRANCISCO RODRIGUES

Uso do Simulador Cooja Contiki em Redes de Sensores Sem Fio (RSSFs) para Sistema de Telemetria em Hidrômetros [Distrito Federal] 2018.

xvi, 37 p., 210 x 297 mm (ENE/FT/UnB, Graduação, Engenharia Elétrica, 2018).

Dissertação de Graduação - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Cooja

2. RSSF's

3. Protocolo

4. Mote

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CUNHA, W.J.F.R. (2018). *Uso do Simulador Cooja Contiki em Redes de Sensores Sem Fio (RSSFs) para Sistema de Telemetria em Hidrômetros*. Dissertação de Graduação, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 37 p.

CESSÃO DE DIREITOS

AUTOR: Webert José Francisco Rodrigues Cunha

TÍTULO: Uso do Simulador Cooja Contiki em Redes de Sensores Sem Fio (RSSFs) para Sistema de Telemetria em Hidrômetros .

GRAU: Graduação em Engenharia Elétrica ANO: 2018

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte dessa Dissertação de Graduação pode ser reproduzida sem autorização por escrito dos autores.

Webert José Francisco Rodrigues Cunha
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

RESUMO

A utilização de um Rede de Sensores Sem Fio (RSSFs), no propósito de monitorar e reduzir as perdas de água, foi simulado um sistema piloto de telemedição em pontos específicos no Campus Darcy Ribeiro da Universidade de Brasília. O sistema registra, com periodicidade horária, o acumulado de consumos em todos os contadores. Esta dissertação consubstancia-se na análise de informações recolhidas no simulador Cooja, com vista à viabilidade de captação de informações dos contadores para que futuramente possa se estabelecer perfis de consumo e perfis de perdas que visem reduzir de forma ativa e passiva as perdas de água. São analisados os perfis de consumo com vista a concretizar ações ativas de redução de perdas. Conclui-se nesta dissertação com a análise dos dados obtidos sobre os nós implementados no simulador.

ABSTRACT

The use of a Network of Wireless Sensors (WSNs), in order to monitoring and reduce water losses, simulated a pilot point telemetry system at the Darcy Ribeiro Campus of the University of Brasilia. The system records, with an hourly period, the accumulated consumption in all counters. This dissertation is based on the analysis of information collected in the simulator, with a view to the feasibility of collecting information from the meters so that future consumption profiles and the profiles can be established in order to actively and passively reduce water losses. The consumption profiles are analyzed in order to implement active actions to reduce losses. It concludes in this dissertation with the analysis of the data obtained on the nodes implemented in the simulator.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS	1
1.2.1	OBJETIVO GERAL	1
1.2.2	OBJETIVOS ESPECÍFICOS	1
1.3	APRESENTAÇÃO DE CONTEÚDO	2
1.3.1	FUNDAMENTAÇÃO TEÓRICA	2
1.3.2	APLICAÇÃO DO TRABALHO	2
1.3.3	AVALIAÇÃO DE DESEMPENHO	2
1.4	ESTRUTURA DO TRABALHO DE CONCLUSÃO DE CURSO	2
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	TELEMETRIA	4
2.2	PROTOCOLOS	5
2.2.1	PROTOCOLO MAC	5
2.2.2	PROTOCOLO DE ROTEAMENTO	6
2.2.3	RPL	8
2.2.4	PROTOCOLO DE TRANSPORTE	11
2.2.5	UDP	11
2.2.6	RPL/UDP	12
2.3	NÓS SENSORES	13
2.3.1	<i>Tmote Sky</i>	13
3	APLICAÇÃO DO TRABALHO	15
3.1	A TELEMEDIÇÃO NO CAMPUS UNIVERSITÁRIO DARCY RIBEIRO	15
3.1.1	DESCRIÇÃO DO CAMPUS UNIVERSITÁRIO DARCY RIBEIRO	15
3.1.2	EQUIPAMENTOS NO CAMPUS UNIVERSITÁRIO DARCY RIBEIRO	16
3.2	SIMULAÇÃO	19
3.2.1	INSTALAÇÃO DE PACOTES	20
3.2.2	INICIALIZANDO O COOJA	20
3.2.3	ADICIONANDO NÓS	21
4	AVALIAÇÃO DE DESEMPENHO	24
4.1	INTRODUÇÃO	24
4.2	PROCESSAMENTO	24
4.2.1	COLETA DE DADOS	26
4.2.2	NOVA TOPOLOGIA	30

5 CONCLUSÃO.....	34
REFERÊNCIAS BIBLIOGRÁFICAS.....	35
APÊNDICES.....	36

LISTA DE FIGURAS

2.1	Tráfego de Comunicação Multiponto-a-Ponto (MP2P)	9
2.2	Tráfego de Comunicação Ponto-a-Ponto (P2P).	10
2.3	Contiki stacks.....	13
3.1	Fotografia aérea do Campus Universitário Darcy Ribeiro (Beto Monteiro, 2011)....	15
3.2	Fotografia aérea da região selecionada (GoogleEarth, 2011)	16
3.3	Fotografia aérea da região selecionada com contadores (GoogleEarth, 2011).....	17
3.4	Fotografia aérea da região selecionada com contadores e concentradores (GoogleEarth, 2011).....	18
3.5	Fotografia aérea da area de cobertura dos concentradores. (GoogleEarth, 2011).....	19
3.6	Cooja Interface	21
3.7	Cooja Selecionando Concentrador	22
3.8	Cooja Criando Concentrador	22
3.9	Cooja Interface Network.....	23
3.10	Cooja Interface Script Editor.....	23
4.1	Cooja Interface Mote output.	24
4.2	Cooja Interface Timeline.	25
4.3	Cooja Interface Sensor Map.	27
4.4	Cooja Gráfico Pacotes Recebidos (por nó).	27
4.5	Cooja Gráfico Pacotes Recebidos (7 nós)	28
4.6	Cooja Gráfico Pacotes Perdidos (7 nós).....	28
4.7	Cooja Gráfico Histórico de Consumo de Energia	29
4.8	Cooja Gráfico Consumo de Energia (por componente)	29
4.9	Cooja Gráfico Ciclo de Trabalho	30
4.10	Cooja Nova Topologia Network.	30
4.11	Cooja Nova Topologia Sensor Map.	31
4.12	Cooja Nova Topologia Gráfico Pacotes Perdidos (12 nós).	31
4.13	Cooja Nova Topologia Gráfico Histórico de Consumo de Energia.....	32
4.14	Cooja Nova Topologia Gráfico Consumo de Energia (por componente)	32
4.15	Cooja Nova Topologia Gráfico Ciclo de Trabalho	33
4.16	Cooja Nova Topologia Gráfico Ciclo de Trabalho	33

LISTA DE TABELAS

4.1	PowerTracker: 13 Motes.....	26
1	PowerTracker: 12 motes	36
2	Nova Topologia Informação dos sensores	36
3	Informação dos sensores	37
4	Média de Consumo de Energia dos Componentes	37
5	Nova Topologia da Média de Consumo de Energia dos Componentes.....	37

LISTA DE SIGLAS

QoS	<i>Quality of Service</i>
LLNs	<i>Low-Power and Lossy Networks</i>
DAG	<i>Directed Acyclic Graph</i>
DODAGs	<i>Destination Oriented DAGs</i>
UDGM	<i>Unit Disk Graph Medium</i>
MAC	<i>Media Access Control</i>
RPL	<i>Routing Protocol for Low Power and Lossy Networks</i>
UDP	<i>User Datagram Protocol</i>
TCP	<i>Transmission Control Protocol</i>
GPS	<i>Global Positioning System</i>
CSMA	<i>Carrier Sense Multiple Access</i>

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A telemetria é já usada ao nível do controle das redes de distribuição, dando informação em tempo real sobre pressões e caudais em pontos específicos da rede, nomeadamente, em nós de fornecimento. O custo atual da tecnologia de Rede de Sensores Sem Fio (RSSF), permitem já utilizar a telemetria de forma massificada na medição dos consumos. Pretende-se aqui analisar os benefícios, o alcance, os custos, as limitações e o problemas na implementação dos sistemas de telemedição.

O Campus Darcy Ribeiro possui em funcionamento um sistema de medição que abrange 58 contadores. Como objetivo, instalar um conjunto de equipamentos no Campus Darcy Ribeiro que recolhem os consumos com periodicidade horária e que totalizam na mesma periodicidade toda a água consumida no Campus. Apura 58 contadores instalados, a informação recolhida no Campus Darcy Ribeiro abre novos caminhos para o estudo e análise dos consumos e do comportamento das redes, configurando-se como o instrumento central deste trabalho e sobre o qual se pretende inferir conclusões que validem ou não a implementação de rádio transmissores nos contadores para coleta de vazão de água.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho de conclusão de curso tem como objetivo o estudo de protocolos de redes, visando a implementação de um sistema de Redes de Sensores Sem Fio (RSSF), utilizando o simulador de rede Cooja (Contiki). Este sistema será responsável pela medição das perdas reais e aparentes em sistemas de distribuição de água através da aplicação e monitorização de sistemas de telemetria.

1.2.2 Objetivos específicos

- Telemetria no Campus
- Apresentar protocolos de redes UDP/RPL
- Simulação da rede no Cooja

1.3 APRESENTAÇÃO DE CONTEÚDO

1.3.1 Fundamentação teórica

Telemetria

Consiste na telemedição para o controle e redução de perdas ativa e passivas em um sistema de distribuição. O sistema permite a medição de pressões e caudais na rede de distribuição, através contadores e todo um conjunto de equipamentos e sistemas de informação de aquisição e processamento de dados de consumo. A utilização de protocolos de comunicação que se adaptem às capacidades técnicas disponibilizadas.

Faz-se a descrição da principal aplicação, limitações e problemas e experiências da sua aplicação. Descreve-se as características principais da telemedição, o tema central desta dissertação. Pondera-se de que forma a telemetria em geral e a telemedição em particular podem contribuir para o controle e redução ativa ou passiva de perdas em sistemas de distribuição. O capítulo conclui-se com a coleta de dados referentes às características técnicas dos sistemas de transmissão de informação correntes em telemetria.

Protocolos

Descreve-se o conjunto de regras que governa a comunicação e troca de dados entre dois ou mais radio transmissores. Os protocolos de estudo serão RPL/UDP, responsáveis pela rede e transmissão respectivamente.

1.3.2 Aplicação do Trabalho

Após a descrição do campus, à sua constituição física. Descreve-se o sistema de telemedição instalado no Campus. Faz-se a descrição do procedimento de análise das informações recolhidas na implementação do sistema de telemedição e apresentam-se os resultados dessa análise por meio de simulação.

1.3.3 Avaliação de Desempenho

Descreve-se os resultados através de todos os dados coletados no decorrer da simulação.

1.4 ESTRUTURA DO TRABALHO DE CONCLUSÃO DE CURSO

Este trabalho possui 5 partes: Introdução, Fundamentação teórica, Aplicação do Trabalho, Avaliação de Desempenho e Conclusão. No capítulo 1 é apresentada a introdução, no capítulo 2

é feito um estudo sobre o referencial teórico, no capítulo 3 é apresentada a proposta de sistema e no capítulo 4 é feita a avaliação do desenvolvimento do sistema, por fim, no capítulo 5 é feita a conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 TELEMETRIA

A telemetria é usada na comunicação entre dois ou mais dispositivos em variadas distâncias, para realizar medição e detecção em pequena ou larga escala. Possibilita também o acesso remoto sem a necessidade de um acesso físico direto.

A palavra telemetria é um *portmanteau* das palavras gregas tele (remote) e metron (para medir). A telemetria é, como o nome indica, sobre a realização de medições remotas. A comunicação máquina-máquina é uma generalização da telemetria que implica comunicação autônoma entre máquinas operadas de forma não humana e é fundamental para o conceito de telemetria. A telemetria é usada para transmitir informações sobre temperatura atual, umidade e vento de estações meteorológicas distantes. A telemetria é usada para transmitir dados de consumo de combustível de caminhões para que o proprietário possa otimizar as rotas do caminhão para economizar custos de combustível e, como consequência, reduzir a poluição. O conceito de comunicação máquina-máquina e telemetria também é usado em distâncias mais curtas. Os marcapassos de hoje (dispositivos que são implantados nos corações das pessoas que tiveram um ataque cardíaco) freqüentemente incluem um dispositivo chamado "bobina de telemetria". Isso permite que um médico monitore a atividade do estimulador sem cirurgia. Em vez disso, o médico usa um dispositivo que cria um campo eletromagnético de baixa potência perto do paciente. A bobina de telemetria reage ao campo elétrico, modulando-o criando um mecanismo de comunicação de baixa potência com o qual a informação pode ser transferida do coração do paciente para o médico. (1)

TRANSMISSÃO AÉREA

Foi escolhido um sistema de rádio bidirecional, pelo fato de poderem receber instruções, diferente do unidirecional que apenas envia informações. Isso facilita a implementação de equipamentos manobráveis que confirmam flexibilidade de operação, através de válvulas telecomandadas, as quais será possível ajustar pressões, caudais ou o circuito da água.

Nos sistemas de rádio bidirecionais, um aparelho receptor emite um impulso para os contadores/concentradores no local pedindo uma resposta. Estes, que se encontram em estado de energia mínima, ligam-se ao receber o sinal e enviam uma resposta ao pedido por via aérea. Mais recentemente os sistemas de rádio bidirecionais são aplicados em sistemas com concentradores. (2)

A implementação de sistema de rádio bidirecionais, ocorrera com o envio de dados em intervalos de tempo programado no próprio nó para o envio aos concentradores. O intervalo de

tempo será pré-determinado de modo que seja preciso e eficiente em termos energéticos. Para constantes envios de dados em pequenos intervalos de tempo, o consumo energético será alto, necessitando de uma bateria um pouco mais robusta. Caso o contrario, temos como alternativa diminuir a potencia do sinal, sendo necessário uma maior proximidade dos nós, mas como os nós são fixos, teríamos que introduzir nós repetidores para possíveis saltos. (3)

O dispositivo utilizado *Tmote Sky* possui antena integrada *onboard* com alcance de 50m *indoor* e um alcance de 125m *outdoor*. Uma opção seria dividir em subsistemas, cada qual com seu concentrador que reuna informação dos contadores das imediações. Caso seja necessário vencer uma grande distância entre o emissor do sinal e a central receptora, estes concentradores podem possuir a função de envio de informação por GSM/GPRS para uma central, tornando o sistema mais oneroso.(4)

A aplicação destes dispositivos sem fio e de baixa potência em campo, permite aumentar significativamente a quantidade de informações coletadas e o número de pontos de controle para uma futura aplicação de acesso remoto.

A parametrização de dados do contador e o eventual fecho de válvulas. Esta última funcionalidade está ainda em estudo no que aos contadores de água diz respeito, não estando esta função ainda comercialmente disponível. Usam se pilhas de menor dimensão já que permanecem na maioria do tempo em modo hibernado com consumo residual de energia eléctrica e permitem leituras a menores distâncias.

As parametrizações mais comuns passam pela indicação de alarmes: fuga de água, desconexão, violação magnética. Note-se contudo que quando ocorrem situações de desconexão do receptor, continua a ser necessário a intervenção física do contador na recolocação em funcionamento, dado ser necessário registrar no dispositivo o volume de água que durante o período de desconexão foi consumido e não registado.

A comunicação bidireccional prevê o uso de chaves de encriptação e de comunicação de hora na comunicação entre dispositivos, precavendo a fraude através de repetição de mensagens antigas. (5)

2.2 PROTOCOLOS

2.2.1 Protocolo MAC

A camada de controle de acesso ao meio físico (MAC), baseia-se em redes de rádio de baixo consumo de energia e baixa taxa de dados do padrão IEEE 802.15.4, com objetivo controlar transmissão de diferentes nós, para que não haja interferência através de mecanismos para determinar quando o meio está ocioso e quando é seguro enviar mensagens. A camada MAC 802.11 oferece suporte para as interferências causadas por colisões pelo múltiplo acesso de sensores. (1)

O uso do CSMA irá detectar ou evitar a colisão, diminuindo a probabilidade de colisão, não a

evitando por completo. Antes de enviar um pacote, cada nó escuta transmissões de outros nós. Se uma transmissão de um outro nó for ouvida, o nó desfará sua própria transmissão por um período aleatório para permitir que o nó transmissor complete sua transmissão.

2.2.2 Protocolo de roteamento

O protocolo de roteamento após a adição de um novo nó, é responsável por estabelecer a conectividade do dispositivo a qualquer outro nó na rede ou determinar se existe um problema de conectividade. O algoritmo de roteamento irá determinar como rotear pacotes de dados na rede e para o dispositivo recém adicionado, dando ao dispositivo a capacidade de executar funções em questão de segundos. Fica a cargo do protocolo de roteamento informar falhas de *link* que inviabiliza o tráfego de dados na rede. (6)

A eficiência da comunicação sem fio entre os nós, vai variar com o tipo de protocolo de roteamento utilizado. O protocolo, no caminho entre a origem e o destino dos dados, irá fornecer os melhores caminhos através da rede, quando os nós não conseguirem se comunicar uns com os outros, sendo necessário um ou mais saltos para a comunicação.

Qualquer algoritmo que calcula rotas para pacotes na rede deve ser capaz de executar cálculos de rota antes de precisar usar a rota. Como esses algoritmos são necessários para reagir a falhas de *link*, informações de uso do *link* e outras propriedades de *link* dinâmico como a informação é distribuída pelo protocolo de roteamento, os algoritmos devem recompor a rota com base no recebimento de novas informações. (7)

A rota é dada pela comunicação criada entre dois nós, que esperam um canal de transmissão de dados limpo para evitar colisões da transmissão de outros nós, estabelecendo um caminho de dados.

O algoritmo de roteamento deve suportar o roteamento do nó, através de restrições impostas pela memória e bateria do dispositivo. O consumo de bateria em alguns nós podem ser mais elevados do que em outros, isto pode ocorrer devido a sistemas altamente escaláveis que geram baixos dados de rede. Infelizmente, os sistemas gerarão um carga pesada na inicialização, já que todos os dados do sensor inicial devem migrar para o nível do controlador. Eles também irão gerar uma carga temporária, mas pesada. (7)

Diagnósticos

O protocolo de roteamento é capaz de ser colocado em um modo "detalhado"ativo. Este modo detalhado fornece informações de comunicação adicionais incluindo, pelo menos, o número total de pacotes roteados enviados e o número de falhas de roteamento (nenhuma rota disponível). Os dados desse diagnóstico devem ser mantidos pelos roteadores continuamente e ser disponível para solicitação a qualquer momento por qualquer outro nó.

Rastreamento de Rotas

Os diagnósticos devem fornecer informações como a qualidade da rota, o número de saltos e as rotas alternativas disponíveis com os custos associados. A qualidade da rota é a medida relativa da "qualidade" da fonte selecionada para a rota de destino em comparação com as rotas alternativas. Este valor composto pode ser medido como uma função de contagem de salto, força do sinal, potência disponível, rotas ativas existentes ou qualquer outro critério considerado pela lista como o diferenciador de custos de rota.

Seleção de rota

A seleção da rota determina a confiabilidade e a qualidade da comunicação entre os dispositivos, otimizando rotas ao longo do tempo e resolvendo quaisquer nuances desenvolvidas no arranque do sistema quando os nós são adicionados de forma assíncrona à rede.

Custo da Rota

O protocolo de roteamento deve possuir uma métrica de qualidade da rota e otimizar a seleção de acordo com essas métricas dentro das restrições estabelecidas para os *links* ao longo das rotas. Essas métricas devem refletir medidas como força do sinal, largura de banda disponível, contagem de saltos, disponibilidade de energia e taxas de erro de comunicação. (8)

Adaptação da Rota

As rotas de comunicação são adaptativas e convergem para a otimização da métrica escolhida (por exemplo, qualidade do sinal, contagem de saltos) no tempo.

Redundância na Rota

A camada de roteamento é configurável para permitir que rotas secundárias e terciárias sejam estabelecidas e usadas após a falha da rota primária.

Tempo de Descoberta da Rota

As aplicações comerciais de missão crítica (por exemplo, fogo, segurança) requerem uma comunicação confiável e uma entrega *end-to-end* garantida de todas as mensagens em tempo hábil. Os tempos limite da camada de aplicação devem ser selecionados de forma criteriosa para cobrir condições anômalas, como pacotes perdidos e/ou descobertas de rotas, mas não devem ser configurados muito grandes para sobrecarregar a resposta da rede. Se a descoberta de rota ocorrer durante o tempo de transmissão de pacotes (roteamento reativo), o tempo de entrega de pacotes

não deve adicionar mais de 120 ms de latência ao tempo de entrega de pacotes.

Preferencia de Rota

O protocolo de roteamento permiti o suporte de rotas preferenciais estáticas configuradas manualmente .

Medidas de Desempenho em Tempo Real

Um nó que transmite um "pedido com resposta esperada" para outro nó deve enviar a mensagem para o destino e receber a resposta em não mais de 120 ms. Este tempo de resposta deve ser alcançável com 5 ou menos saltos em cada direção. Este requisito pressupõe um período de repouso da rede e um tempo de resposta insignificante no nó de destino.(9)

Roteamento Priorizado

A priorização de roteamento de pacotes de rede deve ser apoiado para assegurar que as aplicações de missão crítica (por exemplo, fogo detecção) não podem ser diferidos enquanto o acesso a aplicações menos críticos a rede. O protocolo de roteamento é capaz de fornecer rotas com características diferentes, também conhecido como Qualidade de Roteamento de serviço (QoS). (9)

2.2.3 RPL

RPL é o protocolo de roteamento IPv6 para Redes de baixa potência e perda (LLNs) com algumas restrições impostas aos nós, como o poder de processamento, memória e energia quando operados por bateria ou limitadores de energia. O RPL é pouco usual em tráfego ponto a ponto (entre dispositivos), muito utilizado em tráfegos ponto a multiponto (um ponto de controle central para um subconjunto de dispositivos) ou multiponto a ponto (um subconjunto de dispositivos para um ponto de controle central), podendo se estender de dezenas a milhares de nós. Especificações gerais deste protocolo de rede de baixa potencia e perda (LLN), pode ser encontrado em RFCs, documentos que descrevem padrões de cada protocolo da internet a serem considerados um padrão, como [RFC5867], [RFC5826], [RFC5673] e [RFC5548]. (10)

A topologia é a base do funcionamento do RPL, este usa a topologia como um Grafo Acíclico Direccionado (“*Directed Acyclic Graph*”-DAG) para criar um ou mais destinos orientados (*Destination Oriented DAGs*) (DODAGs), que por sua vez podem estar associados a uma ou mais Instâncias RPL “*RPL Instances*”. Cada RPL *Instance* possui uma topologia única, identificada pelo seu próprio ID (RPLInstanceID). Cada nó tem o seu próprio *rank* e para além disso, cada instância RPL é construída através do seu próprio conjunto de diferentes requisitos, das suas DAGs.(11)

Fluxos de tráfego suportados por RPL

O RPL suporta três fluxos de tráfego básicos: multiponto a ponto (MP2P) ponto-a-multiponto (P2MP) e ponto-a-ponto (P2P).

Tráfego Multiponto a Ponto

O tráfego de comunicação Multiponto-a-ponto (MP2P), ocorre quando vários nós enviam dados para um único nó, chamado de nó concentrador. Para redes com mais de um concentrador, deve-se especificar para qual concentrador os dados devem ser enviados, caso contrário, a rede pode optar por enviar para o concentrador mais próximo do remetente. Isso permite uma maior eficiência na coleta de dados na rede.

Nesta rede de comunicação, o concentrador anuncia a presença enviando mensagens de transmissão repetidas a um remetente para obter o número de saltos e a distância, os nós vizinhos ouvem as transmissões e também enviam uma resposta com suas respectivas distâncias. Assim é possível mensurar a distância e a quantidade de saltos se encontram os transmissores do concentrador. Desta forma todos os nós saberão o número de saltos que se encontram do concentrador e qual dos vizinhos está mais perto do concentrador para o envio de pacotes. Este sistema de contagem de saltos apesar de simples, pode não ser o mais eficiente. Um nó com menor número de saltos do concentrador, pode estar localizado em um lugar com a cobertura de rádio muito ruim, diferente de um nó que apresenta um maior número de saltos do concentrador, localizado em um lugar com uma cobertura de rádio muito boa. Neste caso deve-se priorizar a melhor cobertura de rádio, por apresentar melhores chances do pacote passar sem retransmissões repetidas. (11)

Esta comunicação pode ser usada para coletar informações de status da rede, opção presente para visualização no simulador Cooja, em que o nó concentrador recebe relatórios periódicos de status dos outros nós, tornando possível obter o desempenho geral da rede. Este tipo de comunicação pode ser observado na Figura 2.1.

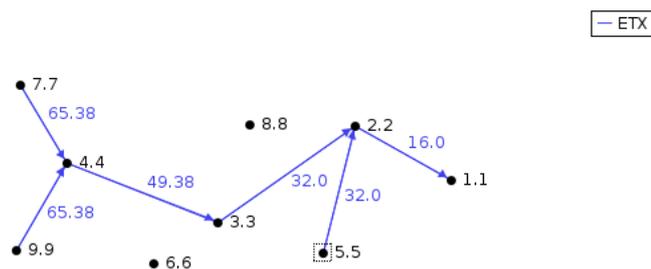


Figura 2.1: Tráfego de Comunicação Multiponto-a-Ponto (MP2P)

MP2P é um fluxo de tráfego dominante em muitas aplicações LLN (RFC5867, RFC5826, RFC5673 e RFC5548). Os destinos de fluxos MP2P são designados à nós que possuem aplicações

significantes, como fornecer conectividade a Internet ou a uma rede de IP privada principal. RPL suporta tráfego MP2P, permitindo que os destinos MP2P sejam alcançados através DODAG roots. (9)

Tráfego de Ponto a Multiponto

O tráfego ponto-a-multiponto apresenta a comunicação de um nó para vários outros nós e possivelmente para todos os outros nós da rede. Para obter uma comunicação confiável entre os nós, é necessário que o protocolo de comunicação seja capaz de detectar as mensagens perdidas e retransmitir-las, até que cada receptor tenha recebido com sucesso. Como meio de tornar a comunicação mais confiável, é usado um mecanismo projetado para redes de rádio de baixa potência, chamado *Trickle*. Ele usa retransmissões periódicas para garantir que as mensagens perdidas sejam retransmitidas. Estas retransmissões podem sobrecarregar o rádio transmissor, para que não ocorra essa sobrecarga, o protocolo utiliza um mecanismo para reduzir o número de mensagens enviadas, atribuindo a cada mensagem um número de sequência, assim o protocolo sabe quais os nós receberam mensagem (se um nó é ouvido enviando um número de sequência antigo, qualquer um dos seus vizinhos pode retransmitir sua última mensagem para o nó com o número de sequência antigo, garantindo que a última mensagem seja conhecida de todos os nós). Este mecanismo, pode ser usado para enviar um comando para um conjunto de nós na rede.

P2MP é um padrão de tráfego exigido por várias Aplicações LLN (RFC5867 ,RFC5826, RFC5673 e RFC5548). RPL suporta o tráfego P2MP usando mecanismo que prevê as rotas para destinos (prefixos, endereços ou grupos de *multicast*) e longe das raízes. Destino anúncios podem atualizar tabelas de roteamento como o DODAG subjacente alterações de topologia.(9)

Tráfego ponto a ponto

O tráfego ponto-a-ponto ocorre quando um nó se comunica com outro nó. Esta comunicação pode envolver outros nós, no entanto, essa comunicação pode ser encaminhada através de uma rede de nós. A Figura 2.2, apresenta a comunicação de dois nós, com dois outros nós envolvidos para o envio de pacotes entre os pontos finais de comunicação.

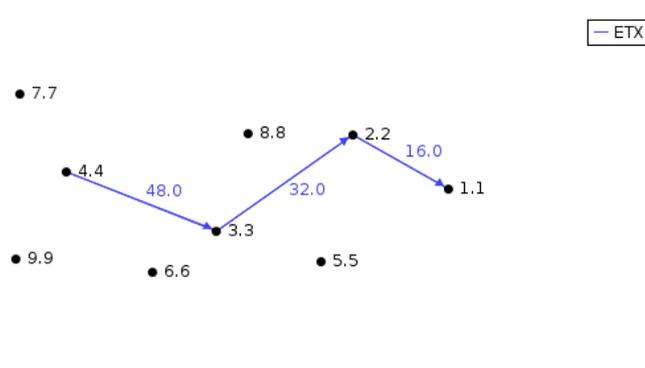


Figura 2.2: Tráfego de Comunicação Ponto-a-Ponto (P2P).

RPL DODAG fornece uma estrutura básica para o tráfego P2P. Para uma rede RPL suportar o tráfego P2P, um *root* deve ser capaz de encaminhar pacotes para um destino. Os nós dentro da rede podem também ter tabelas de roteamento para destinos. Um pacote flui para um *root* até atingir um antecessor que tenha uma rota conhecida para o destino. No caso mais forçado (quando os nós não podem armazenar rotas), esse comum ao antecessor pode ser o DODAG root. Em outros casos, pode ser um nó mais perto da fonte e do destino.(9)

2.2.4 Protocolo de Transporte

A RSSF fornece uma maneira de obter pacotes (unidade convenientes de dados para roteadores) de qualquer nó para um ou mais nós. No entanto os protocolos de rede não garantem a entrega de um pacote, pelo fato de existir a possibilidade dele se perder podendo chegar após outros enviados, serem distorcidos ou simplesmente não serem enviados. Alguns nós incorporam protocolos de transporte que usam a internet para transportar as informações responsáveis pela da aplicação, verificação, correção ou recuperação de erros. Os protocolos de transporte residem na parte superior do protocolo IP. As aplicações não usam IP diretamente, mas usam os protocolos de transporte para se comunicar uns com os outros. (12)

2.2.5 UDP

O *User Datagram Protocol* (UDP) é definido para disponibilizar a comunicação em uma rede nós com comutação de pacotes denominados datagram, pressupondo como protocolo subjacente o Protocolo de Internet (IP), permitindo que a aplicação envie um datagrama encapsulado num pacote IPv4 ou IPv6, e então enviado ao destino.(6)

Os protocolos de transporte mais utilizados são o protocolo de datagrama de usuário (UDP) e o protocolo de controle de transporte (TCP), ambos presentes na pilha de protocolos IP. No UDP, a unidade básica de transporte é chamada de datagrama de dados. Quando discutimos datagramas, estamos nos referindo a um datagrama UDP. A unidade básica TCP de transformação é chamada de segmento.

Por ser um protocolo de baixa complexidade, é usado para transmissão de mensagens mais simples para outros programas com mínimo de mecanismos de protocolo possível, adequado para aplicações que exigem seus próprios mecanismos de roteamento, em que a transação é orientada, mas a entrega e a proteção duplicada não são garantidas. Para aplicações que exigem um maior grau de complexidade, com entrega e fluxo de dados confiáveis, deve-se utilizar o Protocolo de Controle de Transmissão (TCP). O UDP também é adequado para aplicações que exigem seus próprios mecanismos de roteamento. No contexto de redes de objetos inteligentes, o UDP tem muitos benefícios. Primeiro, o UDP tem uma sobrecarga muito baixa para a lógica do protocolo. Isso significa que tanto as transmissões de pacotes quanto as recepções consomem menos energia, e cada pacote tem mais espaço para os dados da camada de aplicação. A simplicidade e a baixa complexidade da lógica do protocolo podem ser vantajosas para sistemas onde a pegada de

memória é superior. Como o protocolo é simples, as implementações normalmente possuem uma pegada de código muito pequena.(13)

A simplicidade do UDP se encaixa muito bem na topologia da RSSF proposta, reportando periodicamente dados. Uma vez que os dados são enviados periodicamente, não importa que os pacotes individuais possam ser perdidos: uma nova leitura de temperatura será enviada o bastante antes. Na proposta, o UDP é adequado ao tráfego com baixas demandas de confiabilidade. É possível fornecer confiabilidade na camada de aplicação, mas isso aumenta a complexidade da aplicação.

O protocolo UDP apresenta algumas desvantagem relacionadas ao projeto. Algumas vezes pode apresentar perdas de pacotes em trânsito, não oferecendo nenhum mecanismo para recuperação desses pacotes. Para o aplicativo recuperar essa perda de pacotes, é necessário aumentar a complexidade das aplicações que exigem confiabilidade. Outra desvantagem seria o fato das RSSF apresentar pequenos tamanhos de pacotes e o UDP não fornece nenhum mecanismo para que as aplicações dividam seus dados em blocos de tamanho apropriado para transmissão. Assim, a aplicação deve descobrir o que é um tamanho de pacote apropriado e ajustar seus pacotes de acordo. Mesmo que a camada IP ofereça suporte para fragmentação e remontagem de pacotes, a fragmentação na camada de IP é frágil, especificamente em redes com perdas. Ao contrário do UDP, o TCP fornece confiabilidade e um mecanismo para limitar automaticamente os tamanhos de pacotes enviados por aplicativos.(14)

Os dois protocolos de transporte mais amplamente utilizados no conjunto de protocolos IP são UDP e TCP. Os protocolos de transporte executam a camada de IP de melhor esforço para fornecer um mecanismo para que as aplicações se comuniquem entre si sem interagir diretamente com a camada de IP. O UDP fornece um serviço de datagramas de melhor esforço onde as aplicações devem fornecer sua própria confiabilidade e controle de fluxo, se necessário. O TCP fornece um fluxo de bytes confiável e reduz a complexidade da aplicação ao custo de um tamanho de cabeçalho maior e uma lógica de protocolo de camada de transporte mais complexa. Para objetos inteligentes, ainda não existe um protocolo de transporte padrão. UDP é leve e simples. Os benefícios do TCP são confiabilidade integrada, controle do tamanho máximo de seus pacotes e interoperabilidade com os sistemas existentes. Os requisitos de aplicação determinam assim a escolha do protocolo de transporte. (1)

2.2.6 RPL/UDP

Neste trabalho de conclusão de curso, UDP é implementado em cima do RPL. Um LLN é composto por um servidor UDP, que aceita pacotes disponíveis e vários clientes UDP, que enviam pacotes periodicamente para o servidor através de um único salto ou múltiplos saltos.

Para ser útil em uma ampla gama de domínios de aplicação LLN, o RPL separa o processamento de pacotes e o encaminhamento a partir do objetivo de otimização de roteamento. Exemplos de tais objetivos incluem minimizar energia, minimizar a latência ou satisfazer restrições.

Uma implementação RPL, em apoio de um aplicativo LLN específico, incluirá a(s) Função(s) Objetiva necessária(s) conforme exigido pela aplicação. A RPL também espera que um mecanismo externo acesse e transporte algumas informações de controle, referidas como "RPL Packet Information", em pacotes de dados. O RPL fornece um mecanismo para divulgar informações sobre a topologia de rede formada dinamicamente. Esta disseminação permite uma configuração mínima nos nós, permitindo que os nós operem principalmente de forma autônoma. Na Figura 2.3 é possível observar as camadas Contiki utilizadas. (5)

<i>Layer</i>	<i>Protocol</i>
Application	IETF CoAP / REST Engine
Transport	UDP
Network	IPv6 / RPL
Adaptation	6LoWPAN
MAC	CSMA / link-layer bursts
Radio Duty Cycling	ContikiMAC
Physical	IEEE 802.15.4

Figura 2.3: Contiki stacks.

2.3 NÓS SENSORES

São componentes responsáveis pelo sensoriamento, processamento e comunicação da rede. Compostos por um microcontrolador, sensores, memória e bateria.

2.3.1 *Tmote Sky*

O *Tmote Sky* é um módulo sem fio de ultra baixa potência para uso em redes de sensores, aplicação de monitoramento e rápida aplicação de prototipagem. *Tmote Sky* alavanca padrões da indústria como o USB e o IEEE 802.15.4 para interoperar perfeitamente com outros dispositivos. Ao usar os padrões da indústria, integrando sensores de umidade, temperatura e luz, e fornecendo interconexão flexível com periféricos, o *Tmote Sky* permite uma ampla gama de aplicativos de rede de malha. O *Tmote Sky* é um substituto para o design bem sucedido do *Telos* da *Moteiv*. O *Tmote Sky* inclui maior desempenho, funcionalidade e expansão. Com o suporte *TinyOS*, a *Tmote* aproveita os novos protocolos sem fio e o movimento do *software* de código aberto. O *Tmote Sky* faz parte de uma linha de módulos com sensores a bordo para aumentar a robustez enquanto diminui o custo e o tamanho do pacote. Este módulo possui uma fonte de energia, um microcontrolador, o transceptor de rádio e um conector de extensão para conectar sensores ou atuadores, tornando a placa um sistema de prototipagem para diferentes aplicações. O sistema usa sensores de umidade, temperatura e luz. (15)

Fonte de Energia

A fonte de energia é uma bateria composta por duas pilhas AA. Qualquer uma das conexões dos terminais da bateria podem fornecer energia para o módulo, podendo ser usada na faixa de operação de 2,1 a 3,6V DC, caso a tensão de entrada exceda essa margem os componentes do módulo podem ser danificados. Usando software e eletrônicos de baixa potência, a vida útil de um única bateria pode durar anos podendo superar a vida útil do sistema.

Microcontrolador

O módulo *Tmote Sky* possui um microcontrolador MSP430 F1611 de baixa potência e um microprocessador com memória interna (10kB de RAM) responsável por conectar o rádio com os sensores e atuadores, temporizadores e *hardware* para conectar dispositivos externos. Devido as as restrições de custo e energia, esses microcontroladores costumam ser significativamente menores do que os microprocessadores utilizados em PCs.

Transceptor de Rádio

O rádio de baixa potência *Chipcon CC2420* compatível com IEEE 802.15.4 e altamente configurável, fornecendo uma comunicação sem fio confiável. A energia de saída à antena é no máximo 1 mW, no entanto, o dispositivo consome 60 mW no total.

Sensores e atuadores

O *Tmote Sky* possui dois conectores de expansão e um par de jumpers onboard que podem ser configurados para que dispositivos adicionais (sensores analógicos, *displays* LCD e periféricos digitais) possam ser controlados pelo módulo *Tmote Sky*, possibilitando interagir com ambiente físico em que são implementados. Os sensores são usados para sentir o ambiente e os atuadores são usados para afetar ou alterar o meio Ambiente.

3 APLICAÇÃO DO TRABALHO

3.1 A TELEMEDIÇÃO NO CAMPUS UNIVERSITÁRIO DARCY RIBEIRO

Com o objetivo de melhorar o sistema de controle de perdas, no Campus Darcy Ribeiro, torna-se necessário atualizar os contadores com a implantação de radiotransmissores, que enviam periodicamente informações sobre o consumo de água nos respectivos contadores. A prefeitura do campus disponibilizou dados referente aos consumos médios e picos de consumo que auxiliarão em um futuro dimensionamento e operação dos sistemas de distribuição e também na estruturação de novos planos tarifários mais segmentados. Informações contida no apêndice do trabalho.

Para uma melhor recepção e qualidade dos dados recolhidos, foi necessário reposicionamento de um dos concentradores e posteriormente ao acrescento de uma terceira unidade de recolha de dados e ao reposicionamento das unidades já existentes. A utilização de um novo equipamento – rádio transmissor – junto ao contador, permite receber dados de pressão e incorporar os resultados obtidos mais facilmente no sistema.

3.1.1 Descrição Do Campus Universitário Darcy Ribeiro

Fundada em 21 de abril de 1962, a UNB teve como primeiro Campus Universitário Darcy Ribeiro. O nome do campus se deve ao fundador e primeiro reitor da universidade, o antropólogo e parlamentar Darcy Ribeiro. Situado no Plano Piloto, o Campus Universitário Darcy Ribeiro conta com cerca de 440 laboratórios, 21 centros, sete decanatos, seis órgãos complementares (Biblioteca Central, Centro de Informática, Editora Universidade de Brasília, Fazenda Água Limpa, UnBTV e Hospital Universitário de Brasília) e seis secretarias. São 109 cursos de graduação e 147 cursos de pós-graduação stricto sensu e 22 especializações lato sensu. (16)



Figura 3.1: Fotografia aérea do Campus Universitário Darcy Ribeiro (Beto Monteiro, 2011).

3.1.2 Equipamentos no Campus Universitário Darcy Ribeiro

Neste trabalho, para RSSF foi utilizada uma arquitetura hierárquica de forma que todos os nós estão organizados em grupos, com nodos dispostos de forma heterogênea junto aos contadores, compondo a área de sensoriamento.

Os nós acoplados aos contadores tem como tarefa, a coleta de pulsos secos de forma de aferir a vazão de água no contador.

Na figura Figura 3.2, apresenta-se a delimitação da área de estudo do Campus através de uma trama a laranja.



Figura 3.2: Fotografia aérea da região selecionada (GoogleEarth, 2011)

Dentro dessa trama, apresenta-se ainda a localização dos vários pontos de entrada de água nas edificações. Importa notar na Figura 3.3 a disposição dos contadores representados em pontos marcados em amarelo, cada bloco podendo possuir mais de um contador. Os contadores aqui representados serão os pontos a implementar os nós transmissores. Devido a imprecisão das coordenadas geográficas dos contadores fornecidas pela prefeitura do campus, foi necessário uma saída de campo com auxílio de um GPS para marcação dos pontos.



Figura 3.3: Fotografia aérea da região selecionada com contadores (GoogleEarth, 2011)

Foram colocados cinco concentradores representados em pontos marcados em verde, que são responsáveis pela captação dos sinais dos vários contadores das imediações. Na Figura 3.4 apresenta-se a localização dos cinco concentradores.

Os concentradores servirão como ponto de acesso onde os novos nós devem objetivar o envio de dados.



Figura 3.4: Fotografia aérea da região selecionada com contadores e concentradores (GoogleEarth, 2011)

A aplicação destes dispositivos sem fio e de baixa potencia em campo, permite aumentar significativamente a quantidade de informações coletadas e o número de pontos de controle para uma futuras aplicações de acesso remoto. Esta topologia, se for bem construída, pode diminuir as falhas de rede, o número de saltos e o número de rotas disponíveis. Nesta topologia espera-se:

- Menor consumo de energia da rede
- Maior tempo de vida útil dos nós
- Tolerância a falhas
- Minimiza o número de colisões na transmissão
- Melhor eficiência na transmissão de dados

Foram utilizados módulos *outdoor* de sensores sem fio *Tmote Sky* nos concentradores, com antena integrada com raio de alcance de 125m. O raio de alcance dos concentradores esta repre-

sentado por uma trama circular verde. Podemos observar na Figura 3.5 que o raio de cobertura abrange todos os contadores das imediações.



Figura 3.5: Fotografia aérea da área de cobertura dos concentradores. (GoogleEarth, 2011)

3.2 SIMULAÇÃO

Nesta dissertação, para o desenvolvimento de uma RSSF, foi necessário o estudo do simulador de rede COOJA e do sistema operacional Contiki. Cooja é um simulador de rede Contiki baseado em java e capaz de emular o *Tmote Sky* dentre outros nós. Trazendo consigo uma série de vantagens, como:

- O código executado pelo nó é exatamente o mesmo do *firmware*, possibilitando fazer o *upload* para os nós físicos.
- Permite a simulação grande e pequenas rede de nós.
- Permite que plataformas de *hardwares* reais sejam emuladas.

- Sua emulação ao nível de *hardware* é lenta porém com uma inspeção muito precisa do sistema.
- A emulação é mais rápida para um nível menos detalhado de grandes redes.

Antes de começar a usar o simulador Cooja, foi necessário visitar a página Contiki e fazer o *download Instant Contiki* e a ferramenta de visualização *VMware*. Para instalar *Instant Contiki* foi necessário um sistema operacional baseado em Linux, o que o torna incompatível com o Sistema Operacional do computador utilizado, logo foi feito o *download* da máquina virtual *VMWare Fusion*, caso o Sistema Operacional fosse *windows*, seria utilizado as máquinas virtuais *VMWare Player* ou *VirtualBox*. Após baixar e descompactar o arquivo "InstanContiki3.0.zip", foi aberto o arquivo executável "Instant_Contiki_Ubuntu_12.04_32-bit", para trabalhar com uma maquina virtual de 32 *bits*, foi instalado todos os pacotes necessários para compilar e executar o ContikiOS.

O computador utilizado para instalação da maquina virtual, possui as seguintes configurações:

- MacBook Pro (13-inch, Mid 2012)
- Sistema Operacional: macOS High Sierra (versão 10.13.3)
- Processador: 2,5 GHz Intel Core i5
- Memória: 4 GB 1600 MHz DDR3
- Gráficos: Intel HD Graphics 4000 1536 MB

3.2.1 Instalação de pacotes

A senha para efetuar o login no *Instant Contiki* é “*user*”. Efetuado o login, na janela de terminal foi dado o seguinte comando para instalar todos os pacotes de suporte para plataforma.

```
sudo apt-get install build-essential binutils-msp430 gcc-msp430 msp430-libc msp430mcu
mspdebug gcc-arm-none-eabi gdb-arm-none-eabi openjdk-8-jdk openjdk-8-jre ant libncurses5-
dev
```

3.2.2 Inicializando o cooja

Feita a instalação, ainda no terminal, é dado o comando “*run_bigmem*” para inicializar o simulador Cooja. O comando “*ant run*” permite a simulação de ate 5 nodos, caso seja necessário simular uma quantidade superior a 5 nodos, que exigirá mais memória, é utilizado o comando “*ant run_bigmem*” como alternativa, isto usa uma combinação de código Java para interface frontal e plataforma específica emuladores para realizar as simulações.

Para dar início a nova simulação, “*File > New Simulation*”. Na caixa de diálogo “*Creat new simulation*” após dar nome a simulação, é selecionado a opção *Unit Disk Graph Medium (UDGM)*, que permite alterar o alcance de transmissão do rádio. Para simulações pesadas que exijam um pouco mais de memória, pode ser selecionado “*No Radio Traffic*”, assim a simulação correrá mais rápida. A caixa de diálogo também dá a opção “*Random start-up delays*” em ms, responsável por variar o intervalo de tempo de início entre cada mote, não permitindo que os nós iniciem ao mesmo tempo. E por último é dada a opção de marcar a caixa “*new random seed on reload*”, semente para gerar números aleatórios. Após criado o ambiente de simulação, aparece uma interface com cinco janelas, conforme observado na Figura 3.6.

1. A janela “*Network*” permite a adicionar, apagar, ou mover fisicamente os nós durante a simulação.
2. A janela “*Simulation control*” permite iniciar, pausar ou reiniciar e limitar a velocidade da simulação.
3. A janela “*Mote Output*” mostra a saída do comando *printf* de todos os nós simulados. A saída pode ser filtrada para mostrar apenas os nós de interesse.
4. A janela “*Timeline*”, pode ser observado em tempo real todos os eventos ocorridos no mote relacionado ao tráfego.
5. A janela “*Notes*” caso seja necessário tomar notas acontecimentos ocorridos durante a simulação.

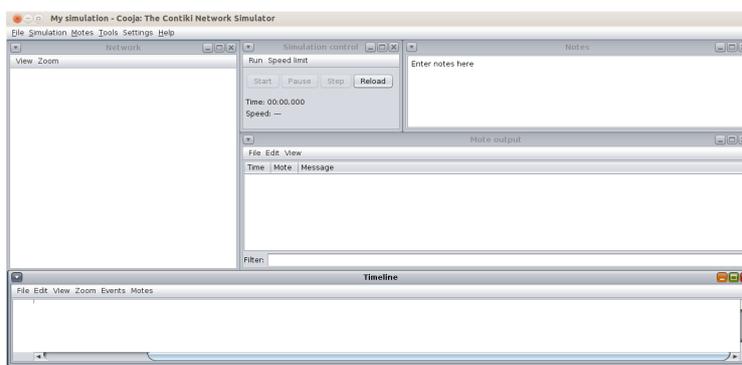


Figura 3.6: Cooja Interface

3.2.3 Adicionando nós

Foram utilizados nós predefinidos disponibilizados no simulador que atendiam ao proposta imposta neste trabalho. Foi criado um pasta nomeada "Tcc" no diretório contiki para armazenar os dados da simulação. Para a adição dos módulos físicos na simulação, foi necessário clicar em “*nós > Add nós > Create New Mote Type*” e selecionar o “*Sky Mote*”, mote escolhido para ser implementado no projeto. Na interface “*Create Mote type*”, o mote é descrito,

`/home/user/contiki/tcc/ipv6/rpl-collect/udp-sink.c`, foi selecionado o arquivo `sink.c`, nó concentrador. Conforme observado na Figura 3.7.

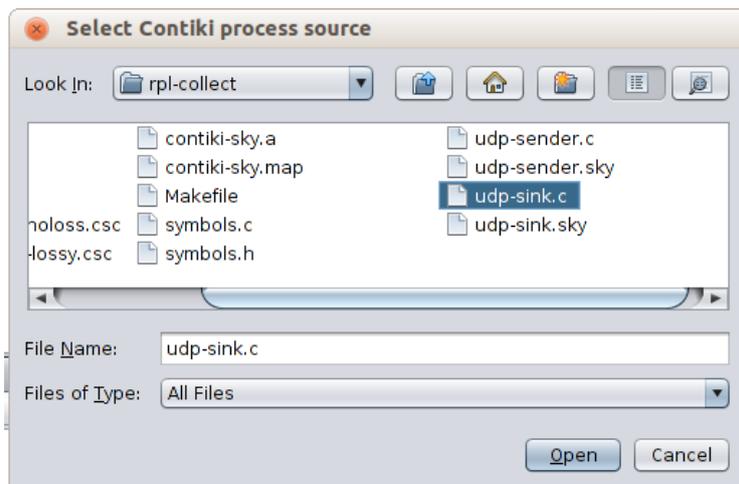


Figura 3.7: Cooja Selecionando Concentrador

Para que seja compilado e criado o nó, o arquivo utilizado tem que ser um arquivo de extensão `.c`, do código fonte de um programa feito em C. Conforme observado na Figura 3.8.

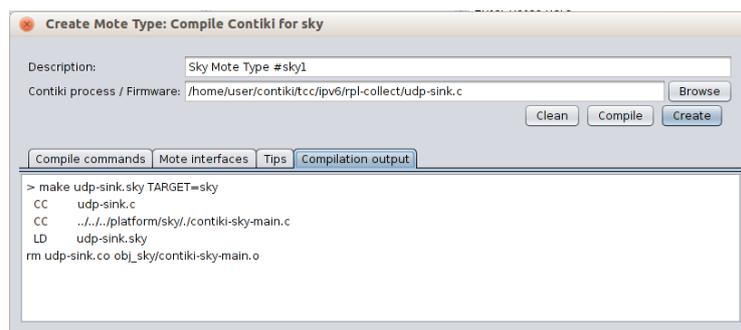


Figura 3.8: Cooja Criando Concentrador

O mesmo procedimento é feito ao selecionar o arquivo `udp-sender.c`, o nó transmissor. Foram adicionados 2 concentradores e 11 transceptores, dispostos de maneira manual seguindo as coordenadas obtidas com auxílio de GPS e respeitando a topologia de rede proposta no trabalho. Na janela “*Network*” é possível os nós que emprega um esquema de cores para diferenciar o tipo dos nós. Foram atribuídas cores verdes aos nós concentradores e cores amarelas aos nós transceptores, conforme observado na Figura 3.9. O raio de cobertura esta delimitado em 125m, representado por uma trama circular verde. A troca de mensagens entre nós que permitem a observação da seleção dos “pais”, como o DODAG é construído e a faixa de transmissão de qualquer nó particular. O *grid* ao fundo apresenta quadros de 10m comprimento.



Figura 3.9: Cooja Interface Network

Para definir um temporizador na simulação, é utilizado “*Simulation Script Editor*”, em um *script* simples foi possível alterar o ponto de *Timeout* para 3600000 milissegundos, observado na Figura 3.10, equivalente a uma hora de simulação, tempo suficiente para captura de dados.

The screenshot shows the "Simulation script editor *active*" window. The editor contains the following JavaScript code:

```

1 /*
2  * Example Contiki test script (JavaScript).
3  * A Contiki test script acts on mote output, such as via
4  * The script may operate on the following variables:
5  * Mote mote, int id, String msg
6  */
7
8 TIMEOUT(3600000);
9
10 while (true) {
11   log.log(time + ":" + id + ":" + msg + "\n");
12   YIELD();
13 }

```

The output window at the bottom shows the following logs:

```

CSMA ContikIMAC, channel check rate 8 Hz, radio channel 26, CCA threshold -45
e link-local IPv6 address fe80:0000:0000:0000:0212:740c:000c:0c0c
'UDP client process' 'collect common process'
nt process started
v6 addresses: aaaa::212:740c:c:c0c
2:740c:c0c
a connection with the server :: local/remote port 8775/5688
ted with address 0.18.116.9.0.9.9.9
ted with address 0.18.116.5.0.5.5.5
2:74:09:00:09:09:09:09 Contiki 3.x started. Node id is set to 9.
2:74:05:00:05:05:05:05 Contiki 3.x started. Node id is set to 5.

```

Figura 3.10: Cooja Interface Script Editor.

4 AVALIAÇÃO DE DESEMPENHO

4.1 INTRODUÇÃO

Na simulação de redes de sensores pode aumentar consideravelmente a velocidade de desenvolvimento e também ser usado para avaliar o consumo de energia, bem como a comunicação e a qualidade da troca de informações. Estes simuladores podem ser usados para facilitar o acesso a resultados finos de experiências correspondentes ao mundo real. Um problema com simuladores é que é difícil mostrar que uma experiência de simulação corresponde bem com um experimento similar do mundo real.

Uma vez que os nós são restritos a recursos, Sistemas Operacionais comuns, pilhas de comunicação e ferramentas de desenvolvimento não podem ser usadas. Isso levou para o desenvolvimento de pequenos Sistemas Operacionais especificamente projetados para rede com restrições de origem. O mais bem conhecido Sistemas Operacionais de controle são Contiki OS, TinyOS e OMNeT++. Todos eles têm ferramentas como simuladores, pilhas de comunicação, são portadas para várias plataformas de *hardware*, são fontes abertas e seu código fonte está disponível na *web*.

4.2 PROCESSAMENTO

Após a simulação ser iniciada, os nós são iniciados com seus respectivos endereços e definidos seus ids e canais. O processo *udp client* é iniciado, como pode ser visto na Figura 4.1.

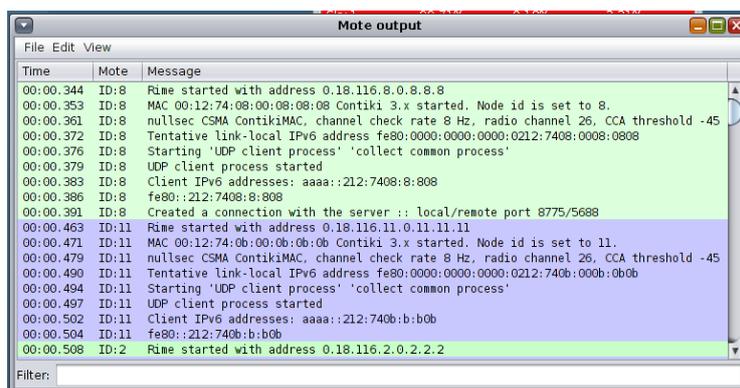


Figura 4.1: Cooja Interface Mote output.

Na interface *Timeline* (Figura 4.2) apresenta os 13 motes simulados, mostrando a comunicação e os eventos do rádio ao longo do tempo.

- Rádio Ligado/Desligado

- Sem cor: radio desligado
- Cinza: radio ligado
- Rádio Rx/Tx
 - Verde: recebeu um pacote
 - Azul: Pacote enviado
 - Vermelho: Interferências (colisões, etc..)

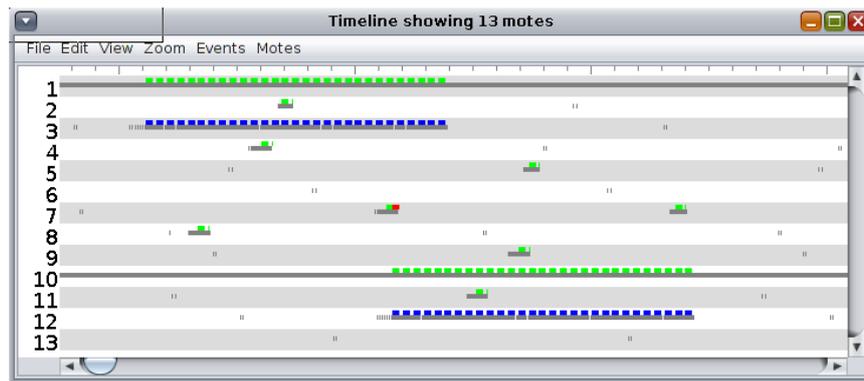


Figura 4.2: Cooja Interface Timeline.

O Cooja também apresenta um *plugin* que fornece a média do ciclo de trabalho dos rádios simulados, chamado *PowerTracker*, é possível observar na tabela 4.1 em porcentagem a energia consumida, tendo como parâmetro o tempo que o rádio ficou ligado, taxa de transmissão e recepção. Os nós de id 1 e 10, são os concentradores que aparecem quase 100% em estado ativo para recepção de dados, enquanto os nós restantes são referente aos transceptores, que ficam em estado ativo apenas quando vão transmitir ou receber dados.

Tabela 4.1: PowerTracker: 13 Motes

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	99.98%	0.03%	0.60%
Sky 2	0.85%	0.08%	0.02%
Sky 3	0.83%	0.08%	0.01%
Sky 4	0.83%	0.08%	0.01%
Sky 5	0.84%	0.08%	0.02%
Sky 6	0.82%	0.08%	0.01%
Sky 7	0.82%	0.06%	0.02%
Sky 8	0.83%	0.09%	0.01%
Sky 9	0.81%	0.06%	0.02%
Sky 10	99.98%	0.03%	0.35%
Sky 11	0.80%	0.07%	0.01%
Sky 12	0.83%	0.08%	0.01%
Sky 13	0.83%	0.09%	0.01%
Average	16.07%	0.07%	0.08%

4.2.1 Coleta de Dados

A Cooja possui ferramentas sofisticadas para coletar dados dos nós simulados. A coleta de dados deve ser realizada a partir do ponto de vista do concentrador, que vai exibir dados de todos nós transmissores das imediações que estabeleceram comunicação com aquele concentrador. Assim a rede de sensores sem fio é dividida em sub-redes, cada qual com um concentrador. Esta topologia diminui o número de saltos e conseqüentemente o número de perdas. Para realização de um estudo efetivo foi selecionado a sub-rede com maior número de nós. A sub-rede selecionada foi a que apresentava o concentrador de id 1, o qual foi utilizado para realizar a coleta de dados dos transmissores de id 2, 3, 4, 5, 6 e 8, conforme apresentado na Figura 4.3.

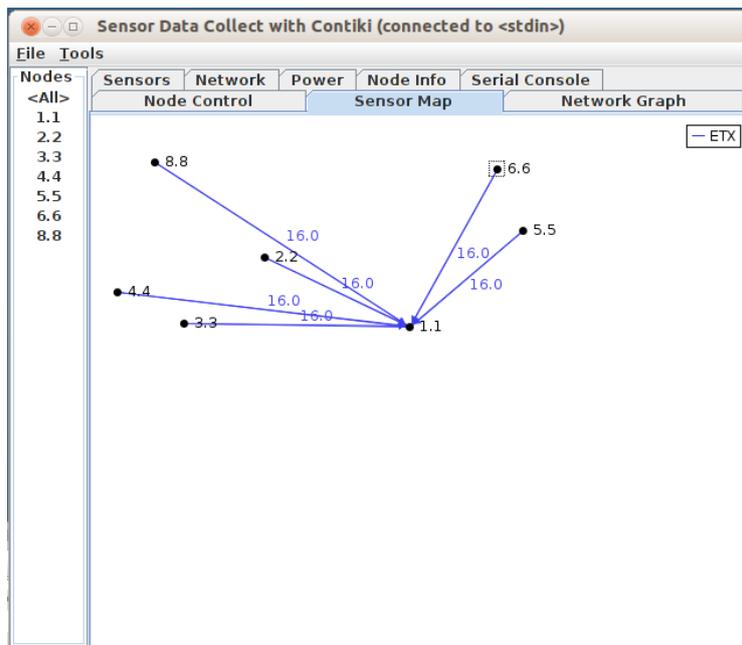


Figura 4.3: Cooja Interface Sensor Map.

Depois de uma hora de simulação, foi possível fazer uma análise dos dados dispostos em seções.

Na seção “*Received (Per Node)*”, mostra a quantidade de pacotes recebidos pelo concentrador dos transmissores a ele conectado, apresentando quantidades iguais de pacotes transmitidos, conforme apresentado na Figura 4.4.

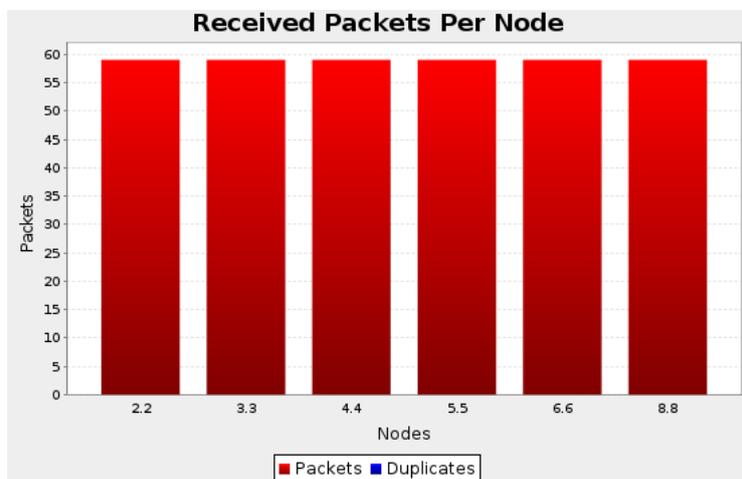


Figura 4.4: Cooja Gráfico Pacotes Recebidos (por nó).

Na seção “*Received (Over Time)*”, mostra um gráfico com número de pacotes enviados pelos 7 nós simulados, totalizando um número de 354 pacotes, estabilizando em 6 pacotes recebidos, conforme apresentado na Figura 4.5.

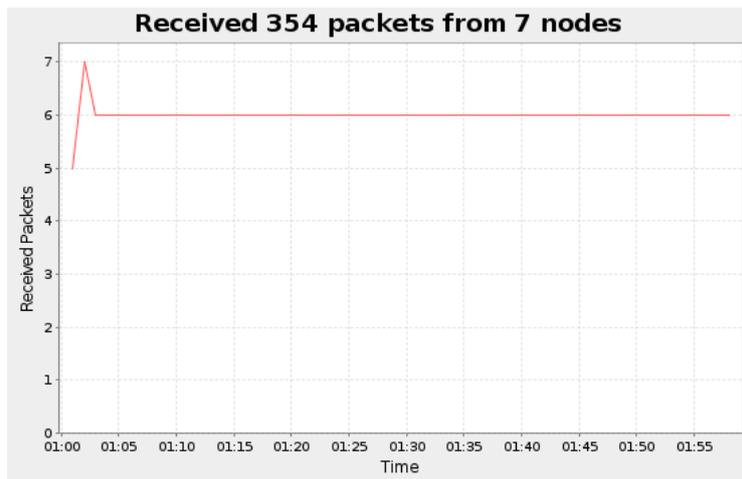


Figura 4.5: Cooja Gráfico Pacotes Recebidos (7 nós)

Na seção “*Lost (Over Time)*”, mostra um gráfico com a estimativa de perda de pacotes durante a simulação. O sistema simulado não apresentou perdas, pelo fato de não apresentar mais de um salto até o concentrador, o número limitado de transmissores em um único concentrador também pode ser um dos fatores determinante para o acontecimento, conforme apresentado na Figura 4.6.

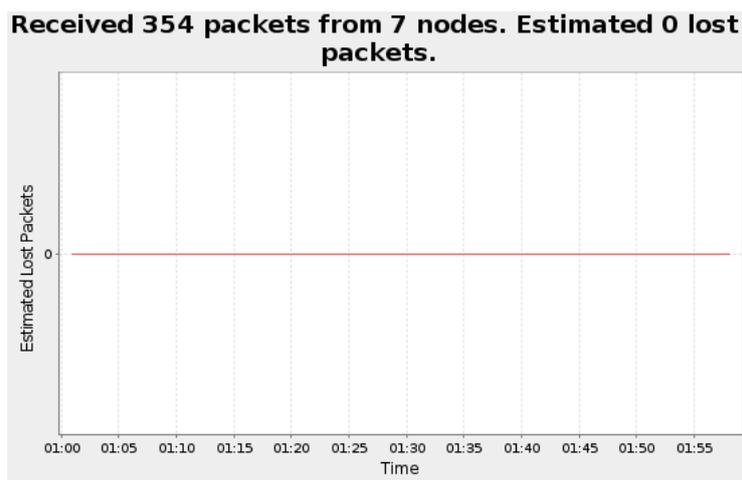


Figura 4.6: Cooja Gráfico Pacotes Perdidos (7 nós).

Na seção “*Power History*”, o gráfico mostra um histórico de consumo de energia de cada nó, observado na Figura 4.7.

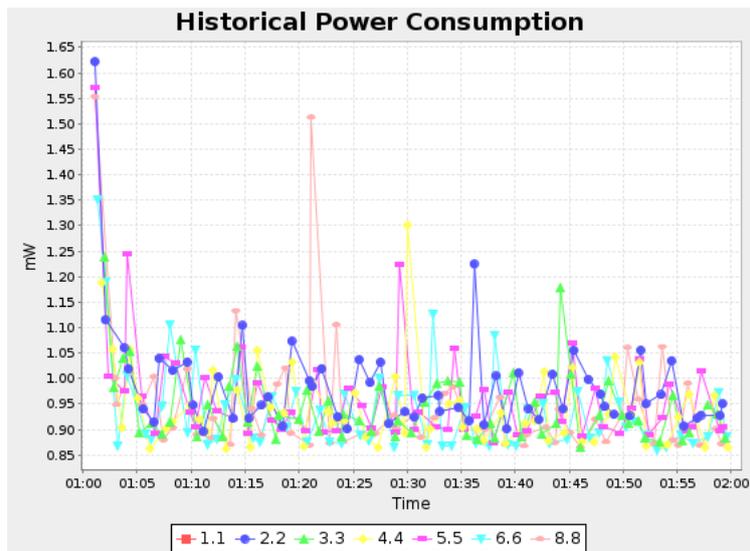


Figura 4.7: Cooja Gráfico Histórico de Consumo de Energia

Na seção “Average Power”, mostra uma média do consumo de energia de cada componente (LPM, CPU, Radio listen e Radio transmit) nos nós, observado na Figura 4.8.

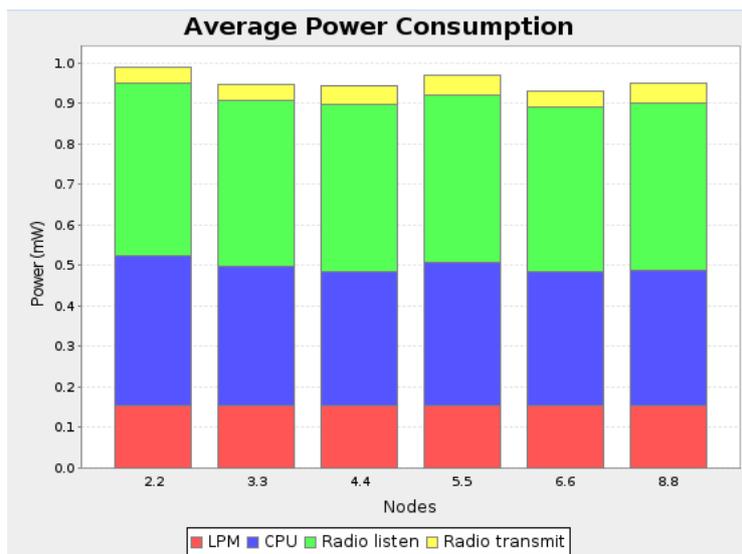


Figura 4.8: Cooja Gráfico Consumo de Energia (por componente)

Na seção “Radio Duty Cycle”, mostra um gráfico com a média do ciclo de trabalho de cada nó, apresentando Radio listen e Radio transmit, conforme observado na Figura 4.9.

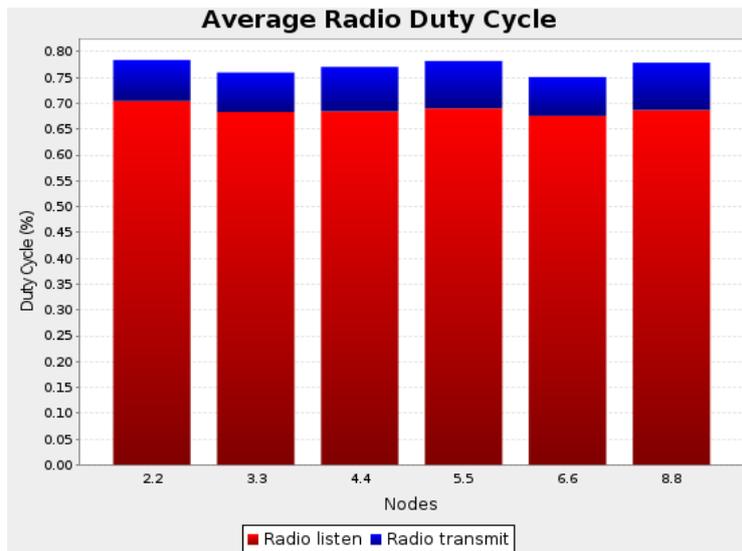


Figura 4.9: Cooja Gráfico Ciclo de Trabalho

4.2.2 Nova Topologia

Apesar do estudo anterior ter sido feito em apenas uma sub-rede, foi adicionado o segundo concentrador para analisar inferências causadas pelo mesmo. Após a retirada do segundo concentrador, três nodos ficaram fora do raio de cobertura do mesmo. Logo foi feito um novo estudo com esta nova topologia, a qual envolve um maior número de saltos no sistema rede.

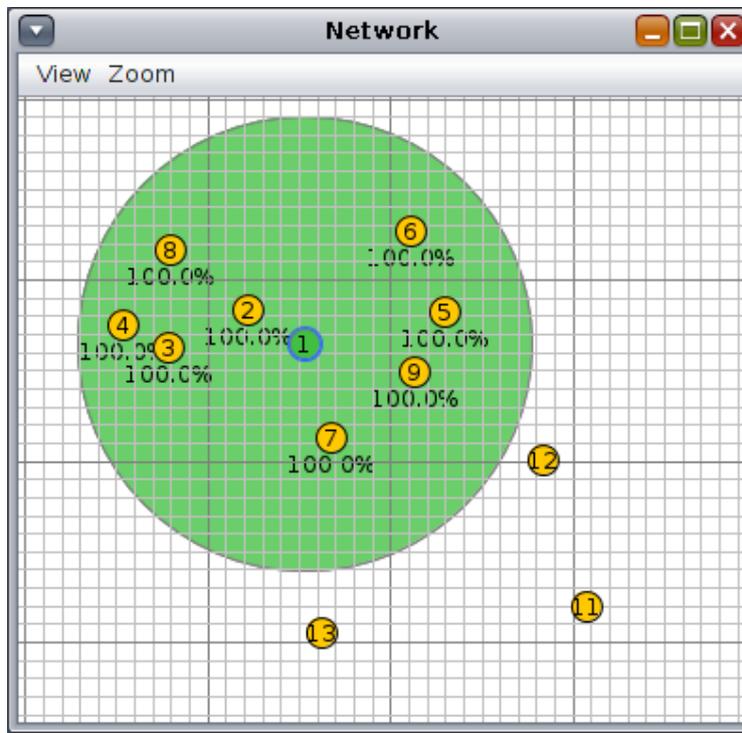


Figura 4.10: Cooja Nova Topologia Network.

Nesta nova topologia aumenta o número de saltos devida a distância de 3 nodos transceptores fora do raio de cobertura. Para realização deste novo estudo efetivo foi selecionado a mesma sub-rede do estudo anterior porém sem o concentrador de id 10. A sub-rede selecionada foi a que apresentava o concentrador de id 1, o qual foi utilizado para realizar a coleta de dados dos transmissores de id 2, 3, 4, 5, 6, 7, 8, 9, 11, 13. Conforme apresentado na figura 4.11, é possível observar os nós de id 11,12 e 13 realizando mais de um salto.

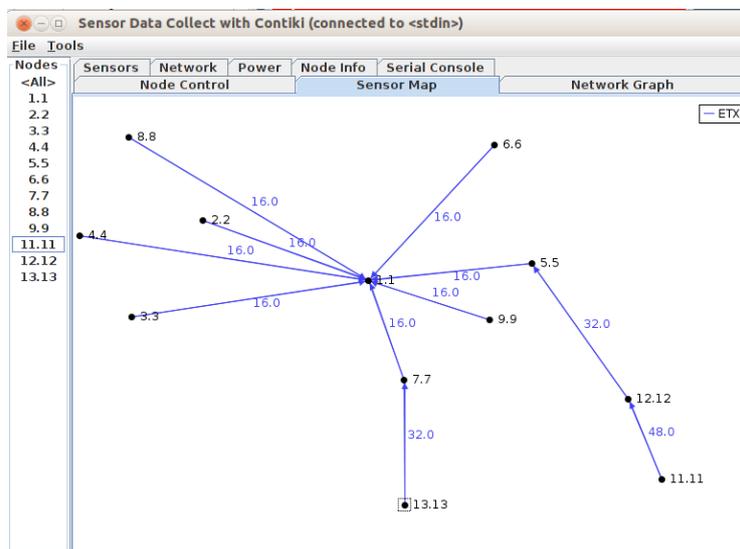


Figura 4.11: Cooja Nova Topologia Sensor Map.

O número de pacotes recebidos pelo concentrador dos transceptores aumentou consideravelmente, de 345 para 649 pacotes, devida a adição de outros 5 nós de id 7, 9, 11, 12, 13, a rede. Conforme apresentado no gráfico da Figura 4.5 e 4.12.

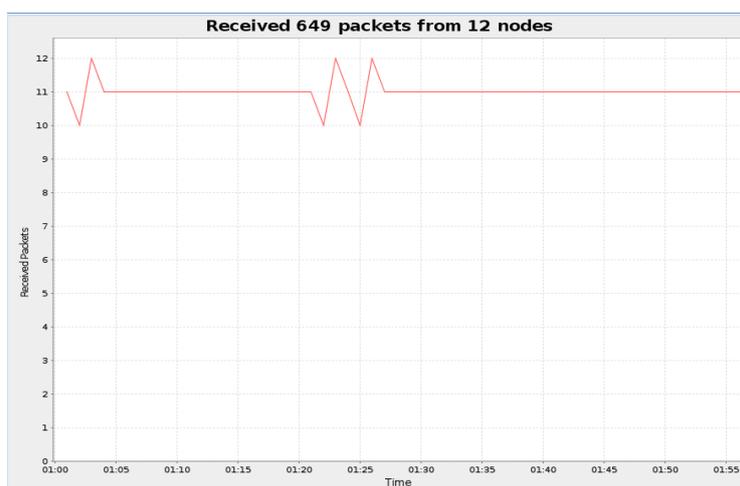


Figura 4.12: Cooja Nova Topologia Gráfico Pacotes Perdidos (12 nós).

Novo gráfico do histórico de consumo de energia de cada nó, onde pode ser observado picos de consumo dos mesmos, conforme o gráfico da Figura 4.13.

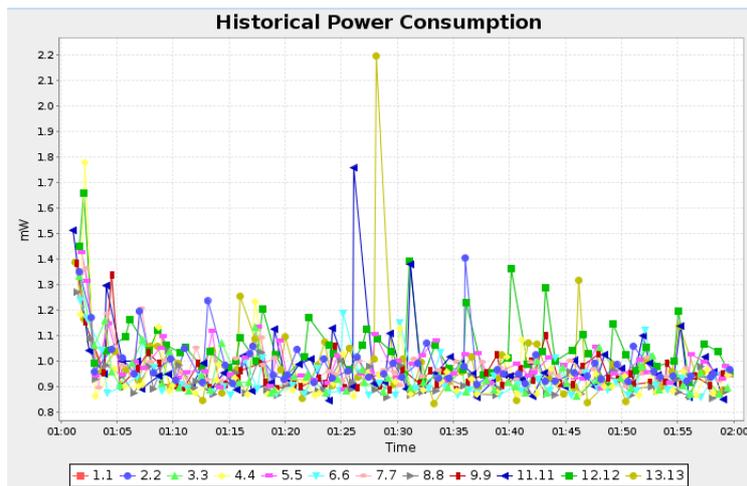


Figura 4.13: Cooja Nova Topologia Gráfico Histórico de Consumo de Energia

A nova média de consumo de energia de cada componente (LPM, CPU, Radio listen e Radio transmit), após a adição de 5 novos nós. Os nós de id 11, 12 e 13 apresentaram um maior consumo na transmissão, devido a distância do nó concentrador de id 1, conforme apresentado no gráfico da Figura 4.14.

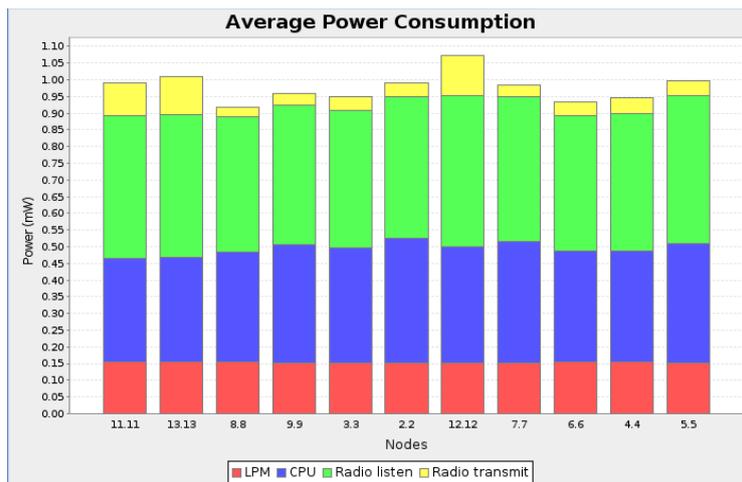


Figura 4.14: Cooja Nova Topologia Gráfico Consumo de Energia (por componente)

A nova média do ciclo de trabalho de cada nó, apresentando maiores ciclos de transmissão nos nós de id 11, 12, e 13, por não estarem no raio de cobertura do concentrador de id 1, conforme apresentado no gráfico da Figura 4.15.

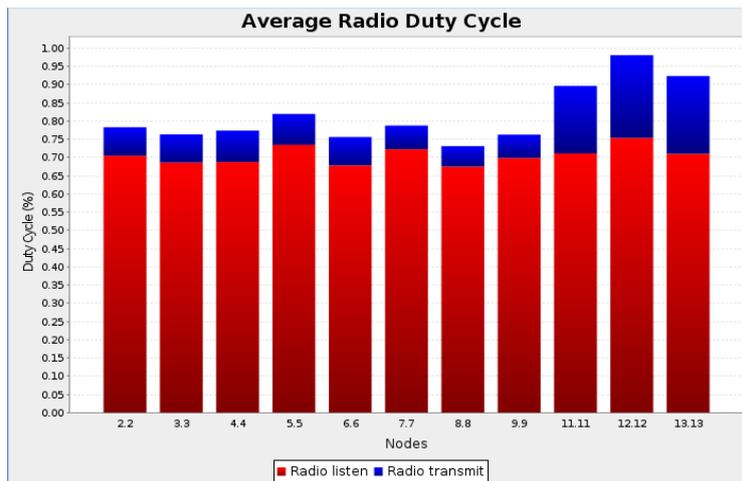


Figura 4.15: Cooja Nova Topologia Gráfico Ciclo de Trabalho

O gráfico "*Network Hops*", apresenta o número de saltos dos nós transceptores para estabelecer contato com o nó concentrador, conforme apresentado na Figura 4.16.

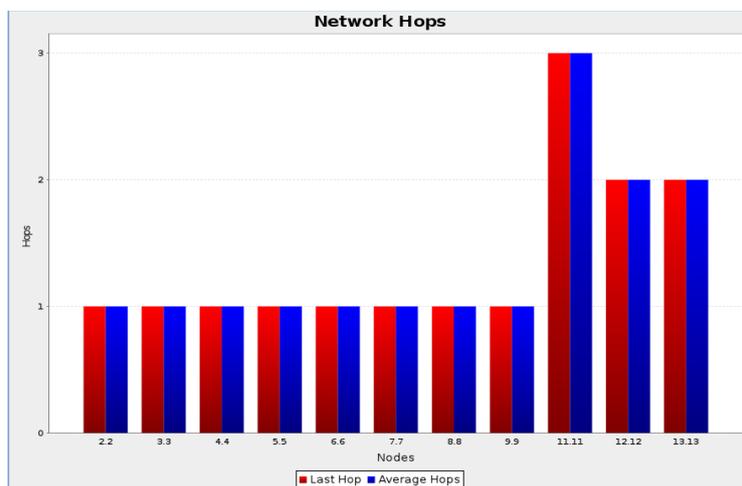


Figura 4.16: Cooja Nova Topologia Gráfico Ciclo de Trabalho

Nesta nova simulação foi possível observar que não existe a necessidade de dois concentradores para obter as informações dos onze transceptores em um sistema sem falha. Em um sistema sujeito a falhas, a falta de um concentrador pode ocasionar nós transceptores "órfãos", prejudicando a coleta de dados naquele sistema.

5 CONCLUSÃO

Neste Trabalho de Conclusão de Curso, foi desenvolvido uma análise de viabilidade de aplicação de uma rede de sensores sem fio para sistema de telemetria em hidrômetros no campus Darcy Ribeiro. A aplicação dessa RSSFs de baixo custo e baixa potência, permite desenvolvimento de soluções complexas e pontuais para o consumo de água no campus.

Foi apresentado o funcionamento do simulador Cooja, o qual se mostrou uma excelente ferramenta para simulação de pequenas RSSF. O simulador também permite que as plataformas de *hardware* reais sejam emuladas. O simulador propõe uma interface clara, de fácil utilização e amigável para monitoramento e controle dos dados coletados na aplicação, possibilita introduzir, retirar ou mover motes, e uma gama de *plugins* que permite a visualização de eventos em tempo real.

Entre os grandes benefícios da utilização do simulador cooja em RSSFs, podem-se citar a capacidade de simular e apresentar em tempo real a comunicação dos nós em uma interface gráfica de fácil entendimento, a praticidade de implementação dos motes para criação da rede e coleta de dados referentes a troca de informações do sistema de comunicação. Por outro lado, foram observadas certas limitações para utilização do simulador Cooja em aplicações específicas na RSSFs. A falta de material disponível dificulta a implementação de alguns protocolos e correção de falhas. A atualização e suporte deixam a desejar, ao tentar instalar pacotes dispostos no site da contiki, as bibliotecas não são encontradas.

De maneira geral, a análise da simulação comprovou que a implementação de uma RSSFs é viável e atende satisfatoriamente as aplicações no sistema de telemetria. Apesar do Campus possuir algumas dezenas de contadores, aos quais possam ser implementados os nodos, as localizações geográficas oferecida pela prefeitura do campus, não são precisas. Por isso, apenas torna-se necessário um mapeamento mais preciso, com uso do GPS. Assim é possível fazer uma simulação em larga escala para o teste de novas aplicações e protocolos.

Em um futuro próximo é expectável que a simulação deste sistema de telemetria aplicado aos contadores do campus, tenha suas funcionalidades aumentada, com monitoramento em tempo real do comportamento das redes de distribuição, permitindo uma gestão e operação destes equipamentos a distância, permitindo a redução dos consumos ilícitos e de fugas de água.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 VASSEUR, J.-P. *Interconnecting Smart Objects with IP: The Next Internet*. Morgan Kaufmann, 2010. ISBN 978-0-12-375165-2. Disponível em: <<https://www.amazon.com/Interconnecting-Smart-Objects-IP-Internet/dp/0123751659?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0123751659>>.
- 2 PACHECO, J. R. B. D. M. *Perdas em Sistemas de Abastecimento Publico de Agua. Uma Nova Abordagem com Base na Telemedicao de Consumos Domesticos*. Dissertação (Mestrado) — FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO, 2010.
- 3 FREEMAN, R. L. *Radio System Design for Telecommunications*. Wiley-IEEE Press, 2007. ISBN 978-0-471-75713-9. Disponível em: <<https://www.amazon.com/Radio-System-Design-Telecommunications-Freeman/dp/0471757136?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0471757136>>.
- 4 SILVA, C. O. da. *SOA-RSSF: Arquitetura Orientada a Servicos (SOA) em Aplicacoes de Rede de Sensores sem Fio (RSSF)*. Dissertação (Mestrado) — UNIVERSIDADE FEDERAL DE SANTA CATARINA, jun. 2008.
- 5 KOVATSCHEK, M.; DUQUENNOY, S.; DUNKELS, A. A low-power CoAP for contiki. In: *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*. [S.l.]: IEEE, 2011.
- 6 POSTEL, J. *User Datagram Protocol*. [S.l.], 1980.
- 7 SHELL, S. D.; PHINNEY, T. *Industrial Routing Requirements in Low-Power and Lossy Networks*. [S.l.], 2009.
- 8 FREIRE, R. E. *Linux Guia do Administrador do Sistema*. [S.l.]: Novatec, 2008. ISBN 9788575221778.
- 9 BRANDT, A.; HUI, J.; KELSEY, R.; LEVIS, P.; PISTER, K.; STRUIK, R.; AND, R. A. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. [S.l.], 2012.
- 10 RPL Basics. <http://anrg.usc.edu/contiki/index.php/RPL_UDP>. Acessado em 18/01/2018.
- 11 SAS, J. V. e Cisco Fellow e Cisco Systems e Zach Shelby e Cedric Chauvenet e W. *Internet Protocol for Smart Objects (IPSO) Alliance*. 2010.
- 12 ISO Transport Protocol specification ISO DP 8073. [S.l.], 1984.
- 13 BORMAN, D. *TCP and UDP over IPv6 Jumbograms*. [S.l.], 1997.
- 14 CERF, V.; KAHN, R. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, Institute of Electrical and Electronics Engineers (IEEE), v. 22, n. 5, p. 637–648, may 1974.
- 15 MOTEIV CORPORATION. *Tmote Sky Ultra low power IEEE 802.15.4 compliant wireless sensor module*. [S.l.], 2006.
- 16 UNB Historia. <<https://www.unb.br/a-unb/historia>>. Acessado em 18/01/2018.

Nesta nova topologia é possível observar a ausência do nó concentrador de id 10, no ciclo de trabalho dos rádios simulados.

Tabela 1: PowerTracker: 12 motes

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	99,98%	0.03%	0.62%
Sky 2	0.84%	0.08%	0.02%
Sky 3	0.84%	0.09%	0.01%
Sky 4	0.83%	0.09%	0.01%
Sky 5	0.89%	0.09%	0.03%
Sky 6	0.83%	0.08%	0.01%
Sky 7	0.86%	0.07%	0.03%
Sky 8	0.82%	0.08%	0.01%
Sky 9	0.84%	0.07%	0.02%
Sky 11	0.92%	0.16%	0.01%
Sky 12	1.01%	0.20%	0.02%
Sky 13	0.93%	0.18%	0.00%
Average	9.12%	0.10%	0.07%

As Tabelas 2 e 3, apresentam Informações de pacotes recebidos, duplicados e perdidos. Métricas como Rtmtric e ETX que auxiliam nos parâmetros de retransmissão e cálculo de rotas.

Tabela 2: Nova Topologia Informação dos sensores

Mote	Received	Dups	Lost	Hops	Rtmtric	ETX
Sky 1	0	0	0	1.000	0.000	0.000
Sky 2	59	0	0	1.000	512.000	16.000
Sky 3	59	0	0	1.000	512.000	16.000
Sky 4	59	0	0	1.000	512.000	16.000
Sky 5	59	0	0	1.000	512.000	16.000
Sky 6	59	0	0	1.000	512.000	16.000
Sky 8	59	0	0	1.000	537.000	16.000
Average	59.000	0	0	1.000	516.282	16.000

Tabela 3: Informação dos sensores

Mote	Received	Dups	Lost	Hops	Rtmetric	ETX
Sky 1	0	0	0	1.000	0.000	0.000
Sky 2	59	0	0	1.000	512.000	16.000
Sky 3	59	0	0	1.000	512.000	16.000
Sky 4	59	0	0	1.000	512.000	16.000
Sky 5	59	0	0	1.000	512.000	16.000
Sky 6	59	0	0	1.000	512.000	16.000
Sky 7	59	0	0	1.000	512.000	16.000
Sky 8	59	0	0	1.000	512.000	16.000
Sky 9	59	0	0	1.000	512.000	16.000
Sky 11	59	0	0	3.000	1025.324	48.000
Sky 12	59	0	0	2.000	768.000	32.000
Sky 13	59	0	0	2.000	785.254	32.000
Average	59.000	0	0	1.364	606.781	16.000

As tabelas 4 e 5 apresentam o consumo de energia de cada componente seguido consumo total dos sensores.

Tabela 4: Média de Consumo de Energia dos Componentes

Mote	CPU Power (mW)	LPM Power (mW)	ListenPower (mW)	Transmit Power (mW)	Power (mW)
Sky 1	0.000	0.000	0.000	0.000	0.000
Sky 2	0.373	0.152	0.423	0.042	0.990
Sky 3	0.343	0.153	0.410	0.041	0.947
Sky 4	0.332	0.153	0.411	0.046	0.942
Sky 5	0.353	0.153	0.414	0.049	0.969
Sky 6	0.332	0.153	0.405	0.040	0.930
Sky 8	0.335	0.153	0.412	0.049	0.949
Average	0.345	0.153	0.413	0.044	0.955

Tabela 5: Nova Topologia da Média de Consumo de Energia dos Componentes

Mote	CPU Power (mW)	LPM Power (mW)	ListenPower (mW)	Transmit Power (mW)	Power (mW)
Sky 1	0.000	0.000	0.000	0.000	0.000
Sky 2	0.375	0.152	0.423	0.041	0.991
Sky 3	0.344	0.153	0.412	0.041	0.950
Sky 4	0.334	0.153	0.413	0.046	0.946
Sky 5	0.358	0.153	0.441	0.045	0.996
Sky 6	0.332	0.153	0.407	0.041	0.934
Sky 7	0.365	0.152	0.434	0.035	0.985
Sky 8	0.331	0.153	0.405	0.030	0.919
Sky 9	0.353	0.153	0.419	0.034	0.959
Sky 11	0.311	0.154	0.427	0.098	0.991
Sky 12	0.346	0.153	0.452	0.121	1.072
Sky 13	0.315	0.154	0.426	0.113	1.008
Average	0.342	0.153	0.424	0.059	0.977