

# Algoritmo de Roteamento de Mensagens Confiável com Controle de Energia Aplicado às Redes Sem Fio Industriais<sup>\*</sup>

Vanessa A. Soria, Max Feldman, Gustavo Cainelli, Tiago Cruz,  
Marcelo Gotz, Ivan Müller

Universidade Federal do Rio Grande do Sul (UFRGS),  
Porto Alegre, Rio Grande do Sul, Brasil.  
(e-mail: [vanessa.soria@ufrgs.br](mailto:vanessa.soria@ufrgs.br), [ivan.muller@ufrgs.br](mailto:ivan.muller@ufrgs.br))

---

**Abstract:** Industrial wireless networks have been widely researched because they demonstrate great flexibility, however must guarantee robustness to meet the requirements of complex applications where communication between devices is deemed critical process. This paper presents an algorithm for establish reliable routes with control over the energy of the nodes. The algorithm is divided into two steps, first the formation of the minimum total length tree between the nodes of the network and in the second step the formation of reliable routes with redundancy for the graphs *uplink* and *downlink*. The proposed algorithm was validated in simulations where different conditions for the state of the nodes are established and the results show the adaptability of the algorithm in establishing the best available routes.

**Resumo:** As redes sem fio industriais (RSFI) têm sido amplamente pesquisadas por demonstrarem grande flexibilidade, por outro lado devem garantir robustez para atender exigências de aplicações complexas onde a comunicação entre os dispositivos é um processo crítico. Este trabalho apresenta um algoritmo para a criação de rotas confiáveis e com controle na energia dos nós. O algoritmo está dividido em duas etapas, primeiro a formação da árvore de comprimento total mínimo entre os nós da rede e na segunda etapa a formação de rotas confiáveis com redundância para os grafos *uplink* e *downlink*. Em cada etapa são considerados parâmetros de estado dos dispositivos. O algoritmo proposto foi validado em simulações onde se estabelecem condições diferentes para o estado dos nós e os resultados mostram a adaptabilidade do algoritmo em estabelecer as melhores rotas disponíveis.

*Keywords:* Industrial wireless networks, routing, reliability, QoS, energy control, graphs.

*Palavras-chaves:* Redes sem fio industriais; roteamento; confiabilidade; QoS; controle energia; grafos.

---

## 1. INTRODUÇÃO

As aplicações das redes sem fio (RSF) vão desde civis até militares como industriais, residenciais, saúde, segurança, entre outras. Uma das principais vantagens dessas redes é que permitem flexibilidade e baixo custo na instalação e manutenção isso porque a mudança e inserção de dispositivos é mais simples além de ter uma redução considerável do custo de cabeamento, no entanto muitas pesquisas também estão sendo encaminhadas a prover uma qualidade de serviço (QoS) cada vez maior para este tipo de redes (Nobre et al., 2015).

Com o passar dos anos RSF têm sido cada vez mais aplicadas na monitorização de ambientes e na busca por ter procedimentos com controle mais dinâmico e preciso. Neste processo de avanços a tecnologia possibilitou o desenvolvimento e integração de dispositivos de sensoriamento e atuação com características de baixo consumo de potên-

cia, pequeno tamanho, autoconfiguração e cada vez mais robustos, esses avanços deram lugar a formação de RSFI, um conceito que vem sendo de interesse e que tem como objetivo incrementar eficiência em aplicações industriais (Chang and Kuo, 2006).

O desempenho dos dispositivos dentro da rede de sensores sem fio irá determinar a segurança e robustez da rede, em geral uma rede sem fio está composta por quatro elementos principais com tarefas bem delimitadas: *dispositivos de campo*, *pontos de acesso*, *gateway*, *gerenciador* conforme visto na Figura 1. Os dispositivos de campo, que correspondem aos sensores e atuadores na indústria, vêm apresentando grande desenvolvimento e podem ter um comportamento autônomo com uma importante capacidade para formar redes e atuar acordo com os diferentes modelos e arquiteturas (Künzel et al., 2012). Embora o comportamento autônomo dos dispositivos de campo seja importante para a robustez da rede, ter um gerenciamento centralizado otimiza seu desempenho em muitas das aplicações. No gerenciamento centralizado se tem uma comunicação e troca constante de dados entre o gerenciador e

---

\* O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

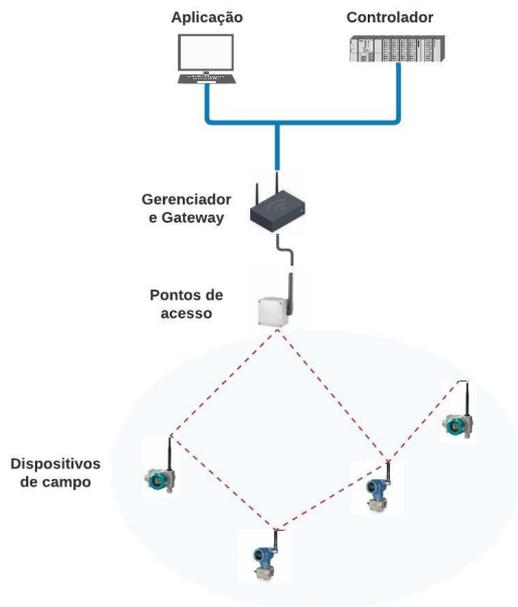


Figura 1. Exemplo de RSFI

os dispositivos finais neste tipo de controle os nós não são capazes de gerar sua própria programação e é o gerenciador o encarregado por essa e outras tarefas como: geração de caminho para o envio dados, controle de adição e saída de dispositivos na rede, maior controle em prevenção de colisões, melhor distribuição de carga e ajuda em mitigar a interferência temporária (Deji et al., 2010).

Além das vantagens das RSFI analisadas em diferentes pesquisas contidas na literatura estas redes podem apresentar problemas a causa das suas restrições, o processo de roteamento em particular consome muitos dos recursos que já são limitados nos dispositivos pelo que é uma área de alto interesse de estudo que procura otimizar o uso de recursos e melhorar o desempenho e tempo de vida da rede para posicionar às RSFI acima das redes cabeadas em muitas das aplicações onde ainda não é possível. Muitos protocolos de roteamento surgiram com diversas propostas: escolha de caminhos ótimos (Han et al., 2011), algoritmo equilibrado em carga (Chu et al., 2009) ou equilibrado em energia (Zhang et al., 2013), assim como também algumas literaturas apresentam um estudo comparativo dos protocolos de roteamento analisando métricas que podem definir o melhor desempenho da rede (Urbina et al., 2015), (Piechowiak et al., 2016).

Este trabalho propõe um algoritmo que apresenta duas etapas as quais são adaptações dos algoritmos em (Dijkstra, 1959) e (Jindong et al., 2009) respectivamente. A primeira etapa define a árvore de comprimento total mínimo sendo que sua construção é feita a partir de uma equação que relaciona: distância entre nós, uma estatística de confiabilidade e o Received Signal Level (RSL) que é uma métrica de potência dada em  $dBm$  que corresponde à intensidade do sinal recebido dos vizinhos do dispositivo. Na segunda etapa do algoritmo são definidas todas as rotas para cada um dos dispositivos considerando redundância, nesta etapa uma nova métrica é calculada baseada no

estado de energia do dispositivo e parâmetros de confiabilidade obtidos dos relatórios de estado dos nós.

As seções deste trabalho são organizadas da seguinte forma: Secção II são apresentadas informações e antecedentes importantes sobre o processo de roteamento por grafo em redes sem fio industriais, na Secção III são apresentados os modelos dos algoritmos usados, equações e métricas são destacadas nessa secção. Na Secção IV é apresentada uma análise sobre o funcionamento do algoritmo e as simulações para validar o modelo e finalmente na Secção V as conclusões do trabalho.

## 2. ROTEAMENTO EM REDES SEM FIOS INDUSTRIAIS

Nas RSFI uma das etapas mais complexas é o roteamento de dados, ele é levado com cuidado por ser redes onde os processos geralmente são muito rigorosos, o tratamento de informação é uma tarefa crítica e a complexidade das aplicações aumenta cada vez, mas com o esse aumento também incrementam as exigências dentro da rede: segurança, confiabilidade na entrega de pacotes, latência e eficiência no consumo da energia são algumas das características em que as RSFI devem ter maior controle para assegurar um ótimo desempenho.

### 2.1 Teoria de Grafos

Nas redes de comunicação sempre procura-se ter uma modelagem o mais perto do sistema real que permita avaliar corretamente métricas ou padrões de comportamento de interesse, neste caminho a representação de uma rede por grafos é muito utilizada independentemente de ser esta uma rede cabeada ou sem fio. Segundo (Deji et al., 2010) um grafo é uma coleção de caminhos que conectam os nós da rede e uma rede modelada formalmente por um grafo é representada como  $G = (V, E)$  onde  $V$  é o conjunto de vértices da rede que representa aos dispositivos e  $E$  é o conjunto de arestas da rede que representa as ligações entre os dispositivos, sendo importante levar em consideração que estas só existem se os dispositivos da ligação são capazes de se comunicar. Em um grafo as setas nas arestas representam o sentido em que a comunicação pode acontecer, é possível encontrar grafos onde as arestas não têm setas o que indica que a comunicação é em ambos os sentidos.

Para nosso posterior uso é importante definir os tipos de grafos no roteamento e seu funcionamento.

- *Grafo broadcast*: estabelece caminhos a partir do grafo original para o envio de pacotes desde o gateway até todos os demais dispositivos da rede.
- *Grafo uplink*: apresenta os caminhos que possibilitam aos dispositivos de campo o envio de dados até o gateway.
- *Grafo downlink*: estabelece as rotas que permitem que o gateway se comunique com um único dispositivo da rede.

### 2.2 Mecanismos de roteamento

Nas RSFI com controle centralizado o gerenciador de rede é o responsável pela criação e armazenamento de

rotas que permitam a comunicação entre os dispositivos, desde o momento em que um dispositivo é agregado na rede ele compartilha com o gerenciador informações que possibilitam a construção e entendimento da topologia da rede. Os seguintes mecanismos são usados pelo padrão WirelessHart e são muito considerados no desenvolvimento de algoritmos de roteamento para RSFL.

- **Roteamento por grafo:** no roteamento por grafo são criados um conjunto de enlaces que vão permitir aos dispositivos na rede estabelecer comunicação e formar rotas para dispositivos distantes. Neste mecanismo é importante fornecer redundância de rotas para melhorar a confiabilidade do processo de roteamento. O gerenciador da rede é o responsável da construção das rotas e de compartilhar as mesmas com os dispositivos, assim em uma rota com nós intermediários eles tem a capacidade de encaminhar corretamente os pacotes de dados até o destino final.
- **Roteamento de origem:** neste mecanismo é estabelecida uma rota entre o dispositivo fonte e destino e não existe redundância de rotas. Este roteamento é geralmente usado só pelo gerenciador e/ou o gateway porque eles têm conhecimento de toda a topologia da rede e sua função principal é a identificação de rotas ou dispositivos em falha. É possível que um dispositivo de campo faça uso deste mecanismo com previa autorização e configuração do gerenciador de rede.

### 3. ALGORITMOS DE ROTEAMENTO PROPOSTOS

O modelo do algoritmo proposto vai definir o grafo *down-link* e *uplink* da rede, eles são construídos em um processo de duas etapas chaves que vão a garantir que as rotas definidas são rotas de comprimento mínimo e que os dispositivos intermediários necessários para o encaminhamento de dados entre o dispositivo fonte e destino têm condições para garantir maior confiabilidade no processo.

#### 3.1 Parte 1: Caminho de comprimento total mínimo

Nesta primeira etapa o objetivo é a construção da árvore de comprimento total mínimo entre todos os dispositivos da rede, no grafo formado a partir desta etapa cada dispositivo da rede tem só uma rota até o gateway e essa rota é a rota com a distância mínima possível. A construção desta árvore é baseada na descrição do problema 1 de Dijkstra (Dijkstra, 1959), após da construção da árvore os dispositivos são dispostos por níveis isso vai trazer certas garantias para o grafo final na etapa dois. Essa distribuição em níveis é feita a partir do grafo Dijkstra obtido.

O pseudocódigo do algoritmo proposto é apresentado no Algoritmo 1 que é baseado em (Dijkstra, 1959).  $G(V, E)$  corresponde ao grafo original formado a partir das informações que os dispositivos compartilham com o gerenciador,  $G_D(V_D, E_D)$  corresponde à árvore de comprimento total mínimo resultante, o conjunto  $SetE_{II}$  são todos as arestas que estão sendo consideradas e que têm probabilidade de formar parte do conjunto  $E_D$ , o conjunto  $SetV_{II}$  os dispositivos que ainda não foram analisados e não formam parte do conjunto  $V_D$ . No começo do programa o gateway representado como  $g$  é adicionado à lista de vértices da

árvore  $V_D$  e as arestas que vão até o gateway desde os pontos de acesso são adicionadas no conjunto  $SetE_{II}$ . Os dispositivos de campo serão representados como  $V_{fd}$  e os pontos de acesso como  $V_{ap}$ . Os seguintes dois passos são seguidos até que os conjuntos  $SetE_{II}$  e  $SetV_{II}$  estejam vazios.

- (1) O conjunto de arestas  $SetE_{II}$  é ordenado de forma crescente segundo (1) e a aresta com o menor valor é removida do conjunto e adicionada ao conjunto  $E_D$  do resultado desta operação o dispositivo destino da aresta é também adicionado ao conjunto  $V_D$ .
- (2) Novamente são considerados todas as arestas que tem como nó fonte o dispositivo recentemente agregado ao conjunto  $V_D$  e como dispositivo destino aos nós que ainda não foram considerados o que significa que formam parte do conjunto  $SetV_{II}$ , esse conjunto de arestas formará parte de  $SetE_{III}$ . É calculada a métrica segundo (1) se o valor é maior que o correspondente no conjunto  $SetE_{II}$  a aresta não é considerada, mas se é menor a aresta substitui a correspondente no conjunto  $SetE_{II}$ . Levando em consideração esta premissa  $SetE_{III}$  e  $SetE_{II}$  são fundidos.

Após a formação completa da árvore é feita a atribuição de níveis para cada um dos dispositivos começando com o gateway como nível 0, os dispositivos a ele conectado com o valor do nível igual a 1 e continuando esse mesmo esquema definindo o valor do nível como o valor do nível de dispositivo fonte mais um.

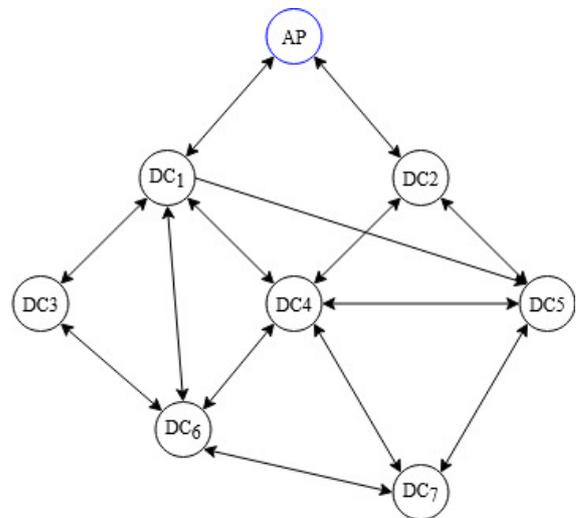


Figura 2. Exemplo de grafo

Do grafo apresentado na Figura 2 foi feita a construção da árvore de comprimento mínimo total representado no grafo da Figura 3 e para finalizar o desenvolvimento desta etapa a Figura 4 apresenta como dispositivos são dispostos por níveis isso vai garantir que todas as rotas estabelecidas para a comunicação com um dispositivo sejam rotas de comprimento mínimo.

A equação que define a métrica usada nesta etapa tem em consideração: o RSL dos vizinhos reportados ao gerenciador por cada um dos dispositivos, a distância em metros entre os dispositivos dispostos na rede e um valor

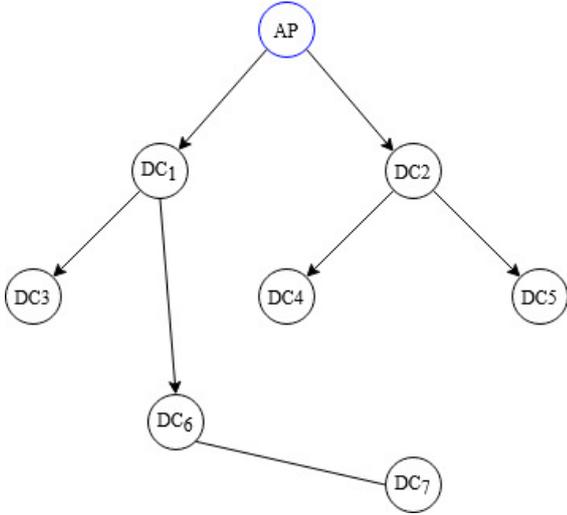


Figura 3. Exemplo: árvore comprimento mínimo

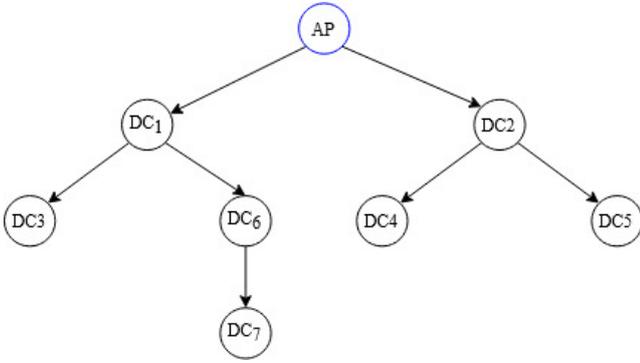


Figura 4. Exemplo: disposição por níveis

de estatística de confiabilidade. Para a definição da equação foram tomadas em consideração os valores do padrão WirelessHart que define um valor mínimo de -85dBm para o valor de RSL e um alcance de comunicação de até 100m (Deji et al., 2010). A equação para determinar a escolha do caminho mínimo entre dois nós é apresentada na Equação 1 e foi gerada a partir de testes que avaliaram a relação entre as métricas.

$$d_{i \rightarrow j} = \frac{D_{i \rightarrow j}}{PR - \frac{100}{RSL_{i \rightarrow j} - 60}} \quad (1)$$

Onde:

$d_{i \rightarrow j}$ : custo da aresta do dispositivo  $i$  para  $j$

$D_{i \rightarrow j}$ : distância entre o dispositivo  $i$  e  $j$

$PR$ : confiabilidade do caminho obtido a partir de métricas do relatório de estado do dispositivo

$RSL_{i \rightarrow j}$ : RSL de  $j$  em  $i$

### 3.2 Parte 2: Formação grafo uplink

Na primeira etapa foi garantido que todos os dispositivos estabeleçam uma rota até o gateway que tem como característica ser a mínima distância possível, no final a etapa 1 vai garantir que todas as rotas redundantes criadas entre um dispositivo fonte e um destino tenham a mesma distância sendo mínimas rotas possíveis. Nesta

**Algorithm 1** Construção árvore de comprimento total mínimo

**In:**  $G(V, E)$

**Out:**  $G_D(V_D, E_D)$

---

Inicialmente  $V_D = g$  e  $SetE_{II} =$  aretas que vão de  $g$  a  $V_{ap}$

- 1: **while** ( $!SetV_{II}.empty()$ ) **do**
- 2:     Ordenar o vetor  $SetE_{II}$  segundo (1)
- 3:     Adiciona  $e_{min}$  o primeiro elemento de  $SetE_{II}$  em  $E_D$
- 4:     Adiciona  $v$  o novo nó em  $V_D$  a partir de  $e_{min}$
- 5:     Armazena em  $SetE_{III}$  arestas que vão de  $v$  a  $SetV_{II}$ .
- 6:     **for** (todos  $e_i$  em  $SetE_{III}$ ) **do**
- 7:         **for** (todos  $e_j$  em  $SetE_{II}$ ) **do**
- 8:             **if** ( $e_i \neq$  todos em  $SetE_{II}$ ) **then**
- 9:                 Adiciona  $e_i$  em  $SetE_{II}$
- 10:             **else if** ( $e_i == e_j$ ) **then**
- 11:                 Compara valores da métrica segundo (1)
- 12:                 Se  $e_i < e_j$  então  $e_j = e_i$
- 13:             **end if**
- 14:         **end for**
- 15:     **end for**
- 16: **end while**
- 17:  $g.setnível(0)$
- 18:  $netdev = g$
- 19: **while** ( $!netdev.empty()$ ) **do**
- 20:     **for** (todos  $v$  em  $netdev$ ) **do**
- 21:         Armazena em  $e_{out}$  aretas com fonte  $v$
- 22:         **for** todas  $e$  em  $e_{out}$  **do**
- 23:              $e.setnível() = 1 + v.getnível()$
- 24:             Armazena em  $netdev_{new}$  o destino de  $e$
- 25:         **end for**
- 26:     **end for**
- 27:      $netdev = netdev_{new}$
- 28: **end while**
- 29: Retorna o  $G_D$  completo

---

segunda etapa vai se construir o grafo *uplink* completo dos dispositivos com redundância de rotas para garantir mínimo duas rotas possíveis entre dois dispositivos, no processo são avaliados parâmetros dos nós para garantir uma maior confiabilidade nas rotas estabelecidas.

O pseudocódigo apresentado no Algoritmo 2 é baseado em *Enhanced Least-Hop First Routing* (ELHFR) (Jindong et al., 2009) e usa o grafo gerado na etapa 1 para a construção dos subgrafos uplink para cada dispositivo. O algoritmo tem como entrada o grafo original  $G(V, E)$  e o grafo criado na primeira etapa  $G(V_D, E_D)$  e gera como resultado os grafos uplink de cada dispositivo  $G_i(V_{ui}, E_{ui})$ , para a obtenção do grafo downlink um processo adicional deve ser seguido mas não será abordado neste trabalho. Para começar o algoritmo é escolhido um nó  $v$  do conjunto  $V_D$  para o qual o grafo  $G_i(V_{ui}, E_{ui})$  será construído, o nó  $v$  com nível  $n$  é adicionado ao conjunto  $V_{ui}$  e os seguintes passos são seguidos até obter o grafo uplink de todos os dispositivos.

- (1) Obter os nós de  $V_D$  com valor de nível  $n - 1$  que tem arestas desde  $v$  e ordenar eles acordo com a métrica dada por (2) com o fim de obter os três melhores candidatos e agregar essas arestas ao conjunto  $E_{ui}$  e os vértices correspondentes em  $V_{ui}$ .

- (2) Criar um vetor  $V_{temp}$  a partir dos dispositivos destino das arestas encontradas em 1 e fazer com cada um deles o processo executado em 1.
- (3) Se o vetor de destinos encontrado no passo 1 tem um só elemento e esse elemento não é o gateway ou um ponto de acesso um dispositivo  $v_n$  em  $V_D$  é escolhido e agregado em  $V_{ui}$  sendo que este será do mesmo nível que  $v$  para garantir sempre a existência de quanto menos duas rotas
- (4) Enquanto  $V_{temp}$  não esteja vazio o processo é repetido desde o passo 1, caso contrário um novo dispositivo do conjunto  $V_D$  é escolhido e o processo de criação do seu grafo uplink é iniciado.

---

**Algorithm 2** Construção rotas confiáveis

**In:**  $G(V, E), G(V_D, E_D)$

**Out:**  $G_i(V_{ui}, E_{ui})$

---

```

1: Criação de vetor de dispositivos já analisados  $S$ 
2: Criação do grafo  $G_i$  para os pontos de acesso onde
    $V_{ui} = g \cup ap_i$ 
3:  $notUsedV = V_D - S$ 
4: while  $!notUsedV.empty()$  do
5:   Escolher um  $v \in notUsedV$ 
6:   Adiciona  $v$  ao conjunto  $S$ 
7:   Adiciona  $v$  ao conjunto  $V_{ui}$ 
8:   Obter  $V_{temp}$  com dispositivos que arestas que vem
   de  $v$ 
9:   Ordenar  $V_{temp}$  segundo (2)
10:  if  $V_{temp}.size() == 1$  then
11:    Obter lista de dispositivos com nível  $n$ 
12:    Ordenar lista de dispositivos obtidos segundo
   (2)
13:    Adicionar a  $V_{temp}$  melhor opção
14:  else if  $V_{temp}.size() > 2$  then
15:    Ordenar  $V_{temp}$  segundo (2)
16:    Deixar em  $V_{temp}$  as duas melhores opções
17:  end if
18:  for todos  $v_v$  em  $V_{temp}$  do
19:    Se existe aresta ente  $v$  e  $v_v$ 
20:    Adiciona a aresta ao conjunto  $E_{ui}$ 
21:    Adiciona o vetor  $v_v$  ao conjunto  $V_{ui}$ 
22:    Repete o processo todo para o dispositivo  $v_v$ 
23:  end for
24:  Atualiza  $notUsedV = V_D - S$ 
25: end while
26: Retorna a lista de grafos  $G_i(V_{ui})$ 

```

---

Nesta segunda parte do algoritmo uma nova métrica é usada para a escolha dos melhores nós disponíveis para a formação das rotas dos grafos, nesta parte as características que são consideradas são parâmetros próprios de cada dispositivo alguns deles correspondem às informações compartilhadas pelos dispositivos ao gerenciador através dos relatórios de estado. Para a análise de energia será considerada o tipo de fonte de energia e o estado da energia do nó sendo que para a especificação de cada um tomaremos a descrição dada pelo padrão WH em (Foundation, 2011). Os parâmetros de confiabilidade são definidos como *confiabilidade de dados* DR e *confiabilidade do caminho* DP estes dois parâmetros são calculados em base a informações compartilhadas nos relatórios de estado. Adicionalmente foram considerados dos valores constantes que ajudam na

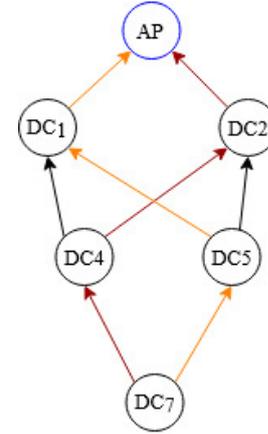


Figura 5. Grafo uplink para o dispositivo 7

validação do modelo dando maior o menor peso a qualquer um dos termos (energia ou confiabilidade).

$$e_i = x_e * \frac{Tsrc_i}{Pst_i + 1} + x_c * \left( \frac{1}{2} - \frac{DR * PR}{DR + PR} \right) \quad (2)$$

Onde:

$Tsrc_i$  : Tipo de fonte de energia usado pelo dispositivo  $i$  considerando: 0. Fonte de poder, 1. Bateria.

$Pst_i$  : Estado da fonte de energia do dispositivo  $i$  considerando: 1. Crítico-Baixo, 2. Recarga baixa, 3. Baixo, 4. Recarga alta, 5. Nominal.

$DR$  : confiabilidade dos dados obtido a partir de métricas do relatório de estado do dispositivo

$PR$  : confiabilidade do caminho obtido a partir de métricas do relatório de estado do dispositivo

$x_e$  : peso estado de energia

$x_c$  : peso estatísticas de confiabilidade A Figura 5 apresenta um grafo uplink como exemplo para perceber como cada um das rotas que podem ser criadas vão sempre garantir com o cumprimento de comprimento mínimo, todas elas vão ter a mesma quantidade de saltos até o gateway.

#### 4. ANÁLISE E SIMULAÇÕES

Na execução do trabalho e para o processo de avaliação se teve como ferramenta base o gerenciador de rede desenvolvido em (Cainelli et al., 2020) onde estão as funções referentes à criação dos dispositivos. Para validar o modelo proposto foram feitas diversas simulações para uma rede com as características propostas na Tabela 1, os dispositivos foram criados e localizados aleatoriamente ao redor da área disposta para rede, e com o fim de validar o funcionamento correto das métricas alguns características envolvidas nas equações propostas foram aleatoriamente estabelecidas nos dispositivos. Os valores dos pesos estabelecidos em (2) foram mudados para estudar como eles podem interferir na construção do grafo e foram propostos cenários onde o tipo de fonte e os dados de confiabilidade foram definidos em valores predeterminados para uma porcentagem  $X$  do número total de dispositivos. Os cenários considerados são apresentados na Tabela 2.

Algumas das métricas consideradas para a avaliação do algoritmo foram propostas também em (Künzel et al., 2017).

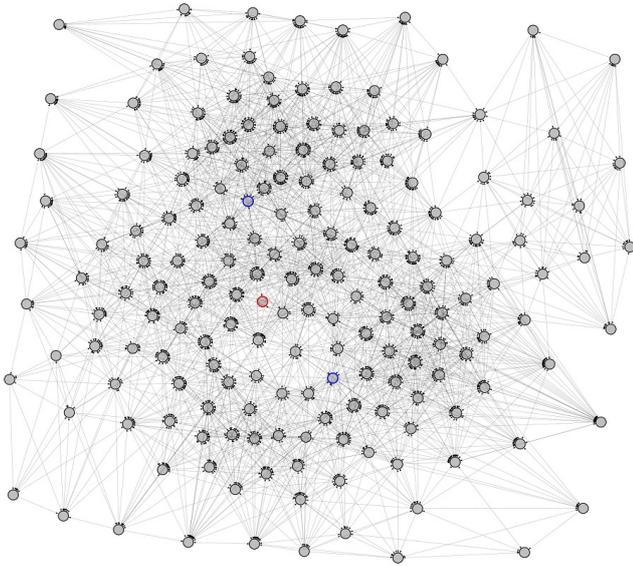


Figura 6. Grafo original rede simulada

- Salto máximo  $h_{max}$ : Corresponde ao maior número de salto entre dois dispositivos.
- Salto médio  $h_{med}$ : Corresponde ao valor médio da quantidade de saltos entre um dispositivo e os pontos de acesso nos subgrafos.
- Dispositivos distantes  $h_{d>4}$ : É a porcentagem de dispositivos que estão localizados a mais de 4 saltos de um ponto de acesso.
- Nós confiáveis  $n_R$ : Porcentagem de dispositivos que tem quanto menos duas arestas de saída no seu correspondente subgrafo uplink.
- RSL média do grafo  $RSL_{med}$ : Corresponde ao valor médio de RSL do grafo.

Os cenários propostos foram avaliados na rede apresentada na Figura 6 a resposta do modelo apresentou características desejadas os resultados obtidos para cada uma das métricas de avaliação são apresentados na Tabela 3, onde a primeira coluna de resultados corresponde aos dados obtidos da rede original sem alterações e com considerações de peso idênticas ( $x_e = 0.5, x_c = 0.5$ ) onde o 50% dos dispositivos tem bateria e outro 50% fonte de energia ilimitada. Lembrando que a geração da rede e informações dos

Tabela 1. Informações da rede

Descrição	Valor
Dimensões da área	450x450
Número de dispositivos	180
Número de pontos de acesso	2
Máximo alcance de comunicação	100m
Mínimo RSL para conexão	-85dBm

Tabela 2. Cenários propostos

Cenários	Considerações		
	$x_e$	$x_c$	RSL no limite
A	0.2	0.8	5% dos FD
B	0.4	0.6	10% dos FD
C	0.8	0.2	30% dos FD

dispositivos foram obtidos de forma aleatória. O número

Tabela 3. Resultados

Métricas	Rede Base	A	B	C
$h_{max}$	7	7	7	7
$h_{med}$	2.81	2.92	3.00	3.1
$h_{d>4}$	10.67%	14.04%	15.73%	17.97%
$n_R$	99.4%	99.4%	99.4%	99.4%
$RSL_{med}$	55.07	56.66	58.29	59.88

de saltos máximo em todos os cenários foi de 7, embora seja um número alto para o tamanho da rede a quantidade de dispositivos alocados neste nível não foi elevado e isso pode ser corroborado com o valor médio de saltos, que para todos os cenários propostos esteve ao redor de 3 sendo que um pequeno aumento é observado dado que o valor RSL de uma quantidade de arestas no grafo base foi também aumentando a cada teste para o valor limite aceitável. A porcentagem de dispositivos distantes a mais de 4 saltos dos pontos de acesso também foi aceitável em geral todos os dispositivos estão localizados a 4 saltos dos pontos de acesso tendo em consideração que cada rota estabelecida até eles é a rota mínima possível e conformada pelos nós mais confiáveis. O modelo é robusto na construção dos subgrafos ele assegurou em todos os cenários que uma porcentagem elevada de dispositivos com rotas redundantes o que proporciona maior confiabilidade para o roteamento de dados.

## 5. CONCLUSÃO

Neste trabalho foram estudados métricas importantes na escolha de nós para a criação de grafos na rede, foram inclusos dados dos relatórios de estados com estatísticas de confiabilidade que são importantes no momento de definir um caminho ótimo para envio de pacotes visando diminuir a probabilidade de falhas na rede. O trabalho apresentado é uma adaptação de dois importantes algoritmos no tratamento com grafos e no desenvolvimento de rotas para RSFI, a proposta poder ser usada para a criação do grado downlink e uplink tendo em consideração que os caminhos definidos são todas as rotas mínimas entre dois dispositivos e com nós de alta confiabilidade no caminho.

Em trabalhos futuros serão adequadas as equações definidas em cada etapa do algoritmo para obter um controle mais fino na construção dos grafos, também vão ser explorados mais parâmetros dentro do funcionamento dos dispositivos que possam prever o comportamento deles, os relatórios de estado e criação de históricos dos dispositivos podem ser uma fonte importante de informação a considerar para o processo de roteamento.

## REFERÊNCIAS

- Cainelli, G., Feldman, M., Cruz, T.R., Müller, I., and Pereira, C.E. (2020). Development of a network manager compatible with wireless hART standard. *XXIII Congresso Brasileiro de Automática (CBA 2020)*, 2, 1–7.
- Chang, R.S. and Kuo, C. (2006). An energy efficient routing mechanism for wireless sensor networks. *20th International Conference on Advanced Information Networking and Applications*, 1, 5–7.

- Chu, Y., Tseng, C., Hung, C., Liao, K., Ouyang, C., Yen, C., Jiang, J., Wang, Y., Tseng, C., and E. Yang (2009). Application of load-balanced tree routing algorithm with dynamic modification to centralized wireless sensor networks. *Sensors*, A247, 1392–1395.
- Deji, C., Mark, N., and Aloysius, M. (2010). Wireless-hart, real-time mesh network for industrial automation. *Springer Publishing Company, Incorporated*, 1.
- Dijkstra, E.W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1, 269–271.
- Foundation, H.C. (2011). Hcf\_spec-183, revision 21.0 common tables specification. *WirelessHart*.
- Han, S., Zhu, X., Mok, A.K., Chen, D., and Nixon, M. (2011). Reliable and real-time communication in industrial wireless mesh networks. *17th IEEE Real-Time and Embedded Technology and Applications Symposium*, 3–11.
- Jindong, Z., Zhenjun, L., and Yaopei, Z. (2009). Elhfr: A graph routing in industrial wireless mesh network. *International Conference on Information and Automation*, 22–25.
- Künzel, G., Noble, B., and Sneddon, I.N. (2012). Ambiente de avaliação de estratégias de roteamento em redes wirelesshart. *M. Eng. thesis, Universidade Federal do Rio Grande do Sul*.
- Künzel, G., Pedroso, G., and Pereira, C.E. (2017). A weighted broadcast routing algorithm for wirelesshart networks. *VII Brazilian Symposium on Computing Systems Engineering*.
- Nobre, M., Silva, I., and Guedes, L. (2015). Routing and scheduling algorithms for wirelesshart networks: A survey. *Sensors*, 15, 9703–9740.
- Piechowiak, M., Zwierzykowski, P., Owczarek, P., and Wasłowicz, M. (2016). Comparative analysis of routing protocols for wireless mesh networks. *10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 1–5.
- Urbina, M., Acosta, T., Christian, V., and Chaffa, G. (2015). Analysis and comparison of routing protocols in wireless sensor networks under 802.15.4. *IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)*, 1–6.
- Zhang, S., Yan, A., and Ma, T. (2013). An energy-balanced graph routing algorithm for wirelesshart networks. *5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2, 557–560.