



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# **Comunicação Cooperativa: Uma Alternativa Para A Resiliência Em Redes Ad Hoc Sem Fio**

Ulisses Rodrigues Afonseca

Tese apresentada como requisito parcial para  
conclusão do Doutorado em Informática

Orientador  
Prof. Dr. Jacir Luiz Bordim

Brasília  
2015

Ficha catalográfica elaborada automaticamente,  
com os dados fornecidos pelo(a) autor(a)

AAF257 Afonseca, Ulisses Rodrigues  
c Comunicação Cooperativa: Uma Alternativa Para A  
Resiliência Em Redes Ad Hoc Sem Fio / Ulisses  
Rodrigues Afonseca; orientador Jacir Luiz Bordim. --  
Brasília, 2015.  
115 p.

Tese (Doutorado - Doutorado em Informática) --  
Universidade de Brasília, 2015.

1. articulações. 2. comunicação cooperativa. 3.  
pontes. 4. redes ad hoc. 5. conectividade. I.  
Bordim, Jacir Luiz, orient. II. Título.



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Comunicação Cooperativa: Uma Alternativa Para A Resiliência Em Redes Ad Hoc Sem Fio

Ulisses Rodrigues Afonseca

Tese apresentada como requisito parcial para  
conclusão do Doutorado em Informática

Prof. Dr. Jacir Luiz Bordim (Orientador)  
CIC/UnB

Prof. Dr. Mário Antônio Ribeiro Dantas  
INE/UFSC

Prof. Dr. Eduardo Adilio Pelinson Alchieri  
CIC/UnB

Prof.<sup>a</sup> Dr.<sup>a</sup> Genáina Nunes Rodrigues  
CIC/UnB

Prof. Dr. Li Weigang  
CIC/UnB

Prof.<sup>a</sup> Dr.<sup>a</sup> Célia Ghedini Ralha  
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 27 de novembro de 2015

# Dedicatória

Aos meus pais Jafet e Hildete, que me ensinaram a importância da família e o caminho da honestidade e perseverança. À minha família por entender minha ausência nos muitos momentos desde que ingressei no doutorado. À minha esposa Juliane, por compartilhar esta jornada.

# Agradecimentos

Ao professor Jacir Luiz Bordim por compartilhar o seu conhecimento e me guiar nesta jornada. Aos professores Marcos Fagundes Caetano, Marcus Vinicius Lamar e André Costa Drummond pelo apoio durante o desenvolvimento da pesquisa. Ao amigo Thiago Fernandes Neves pelas inestimáveis contribuições no desenvolvimento deste trabalho. Ao amigo Thiago Peixoto dos Reis por compartilhar as várias horas de viagem e pelo incentivo. Aos colegas de estudo, Waldeyr Mendes Cordeiro da Silva, Lucas de Melo e André Luiz Peron Martins Lanna por compartilharem o aprendizado. Em especial aos meus familiares Jafet Rodrigues Afonseca, Hildete Lemos Afonseca, Juliane Xavier Dos Santos, Deborah Rodrigues Afonseca, Etiene Rodrigues Afonseca, Suelen Rodrigues Afonseca, Hailton Bezerra dos Santos, Jurandi Alves dos Santos, Eliéce Xavier dos Santos, Eliane Xavier dos Santos e Antônio Xavier dos Santos pelo apoio e suporte. À toda equipe do Departamento de Computação, aos técnicos pela inestimável ajuda nas tarefas administrativas e aos professores pelos ensinamentos.

# Resumo

As redes *ad hoc* sem fio constituem uma alternativa de baixo custo e de fácil implementação para o suporte a ações temporárias ou emergenciais em locais onde uma infraestrutura de comunicação convencional encontra-se indisponível. Apesar da sua capacidade de se adaptar a alterações na topologia, manter ou mesmo reforçar a conectividade da rede torna-se fundamental para o sucesso de diversas aplicações, em particular as de resgate e salvamento. Diversos trabalhos abordam esta questão no intuito de prover mecanismos para mitigar eventuais falhas na topologia que possam inviabilizar a conectividade da rede. Esta tese demonstra que a Comunicação Cooperativa pode ser efetivamente utilizada para aumentar a conectividade das redes *ad hoc* sem fio. Em uma CC, os nós portadores de uma única antena se beneficiam da diversidade espacial de forma semelhante a um sistema *Multiple-Input Multiple-Output*. Esta técnica permite ampliar o raio de comunicação dos nós devido à sua característica de combater o desvanecimento de sinal e os enlaces cooperativos podem ser empregados para reforçar a conectividade da rede. Contudo, as soluções existentes são, em sua maioria, centralizadas e apresentam um alto custo computacional, o que as tornam inviáveis no contexto da recuperação de conectividade. Ao aplicar o conhecimento global da topologia, o melhor enlace cooperativo, entre componentes isolados da rede, tem um custo computacional de  $O(|V|^2 \cdot \Lambda)$ , onde  $|V|$  corresponde ao número de nós da rede e  $\Lambda$  equivale ao custo para selecionar os nós auxiliares em um enlace cooperativo. Esta tese propõe duas soluções distribuídas que aplicam conhecimento localizado da topologia e permitem aumentar a conectividade da rede com objetivo de reforçar a conectividade por meio da utilização de enlaces cooperativos. As articulações, também conhecidas como nós críticos ou vértices de corte, são utilizadas para identificar possíveis pontos de segmentação da rede. A primeira solução requer  $O(\Delta(G)^2 \cdot \Lambda)$  unidades de tempo para determinar, a partir das articulações, o melhor enlace cooperativo localizado a dois saltos, onde  $\Delta(G)$  representa o grau da rede. Na segunda solução, os vizinhos das articulações computam o enlace cooperativo de reserva com um custo total de  $O(\Delta(G) \cdot \Lambda)$ . Em caso de segmentação da rede, estas soluções permitem que enlaces cooperativos sejam ativados para reestabelecer a conectividade. Da mesma forma, os enlaces cooperativos podem ser exploradas para manter a biconectivi-

dade da rede e permitir o balanceamento de carga. Esta tese também apresenta uma alternativa distribuída para a localização de articulações. Comparado com soluções existentes, a alternativa proposta utiliza um volume de mensagens significativamente inferior. Uma abordagem incremental, e diferente das existentes na literatura, é utilizada para determinar as pontes e os componentes biconectados.

**Palavras-chave:** articulação, comunicação cooperativa, ponte, redes *ad hoc*, conectividade

# Abstract

Wireless ad hoc networks are low cost and easy to deploy alternative to support temporary or emergency tasks in places where conventional communications infrastructure are unavailable. Despite its ability to adapt to topology changes, maintain or even enhance network connectivity is fundamental to the success of many applications, in particular those of search and rescue. Several papers address this issue in order to provide mechanisms to mitigate eventual network faults that might preclude network connectivity. This thesis shows that Cooperative Communication (CC) can be effectively used to increase the connectivity of ad hoc wireless networks and restore network connectivity when needed. In CC, nodes with a single antenna benefit from spatial diversity similarly to Multiple-Input Multiple-Output (MIMO) systems. This technique allows the nodes to extend their communication range due to its capability of reducing signal fading, thus making CC a suitable alternative for enhancing network connectivity. However, existing solutions are usually centralized and have a high computational cost, making them unviable in the context of connectivity recovery. When full network topology is known beforehand, the best cooperative link between isolated network components can be computed in  $O(|V|^2 \cdot \Lambda)$  time, where  $|V|$  corresponds to the number of nodes network and  $\Lambda$  is the cost to select auxiliary nodes to form the cooperative link. This thesis proposes two distributed solutions that apply localized topology knowledge to increase and restore connectivity through the use of cooperative links. Articulations, also known as critical nodes or cut vertices, are used to identify potential network segmentation points. The first solution requires  $O(\Delta(G)^2 \cdot \Lambda)$  time units to determine, based on the articulation vicinity, the best cooperative link between clusters of nodes using two hop localized information, where  $\Delta(G)$  represents the network degree. In the second solution, the articulation neighboring computes the backup cooperative link with a total cost of  $O(\Delta(G) \cdot \Lambda)$  time. In case of network segmentation, the proposed solutions activate the computed cooperative links to restore connectivity. Also, the cooperative links can be exploited to obtain a biconnected network. This thesis also presents a distributed alternative to locate articulations. Compared to existing solutions, the proposed alternative uses a significantly lower volume of messages. Furthermore, by using an incremental approach, it is possible to determine



bridges and biconnected components.

**Keywords:** articulation, ad hoc networks, bridge, connectivity recovering, cooperative communication

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Definição do Problema . . . . .	4
1.2	Objetivos . . . . .	5
1.3	Metodologia . . . . .	5
1.4	Contribuições . . . . .	6
1.5	Organização do Trabalho . . . . .	6
<b>2</b>	<b>Articulações e Pontes em Redes <i>Ad Hoc</i> Sem Fio</b>	<b>8</b>
2.1	Articulações e Pontes . . . . .	9
2.2	Localização de Articulações e Pontes em Redes <i>Ad Hoc</i> Sem Fio . . . . .	14
2.3	Mitigando os Efeitos da Presença de Articulações . . . . .	22
2.4	Considerações . . . . .	27
<b>3</b>	<b>Comunicação Cooperativa</b>	<b>29</b>
3.1	Esquemas de Comunicação Cooperativa . . . . .	30
3.2	Modelo de Comunicação Cooperativa . . . . .	35
3.3	Incremento de Conectividade Utilizando Comunicação Cooperativa . . . . .	38
3.4	Considerações . . . . .	45
<b>4</b>	<b>Localização de Elementos Críticos e Componentes Biconectados</b>	<b>47</b>
4.1	Algoritmo Distribuído Para Localização de Articulações . . . . .	48
4.2	Algoritmo Distribuído Para Localização de Pontes . . . . .	53
4.3	Algoritmo Distribuído Para Definição de Componentes Biconectados . . . . .	57
4.4	Resultados Empíricos . . . . .	61
4.5	Síntese e Discussão . . . . .	66
<b>5</b>	<b>Recuperação de Conectividade</b>	<b>69</b>
5.1	Solução Distribuída e Localizada a 2 Saltos . . . . .	70
5.1.1	Detalhamento da Solução . . . . .	70
5.1.2	Resultados Empíricos . . . . .	75

5.2	Solução Distribuída e Localizada a 1 Salto . . . . .	86
5.2.1	Detalhamento da Solução . . . . .	87
5.2.2	Resultados Empíricos . . . . .	93
5.3	Síntese e Discussão . . . . .	102
<b>6</b>	<b>Conclusão</b>	<b>105</b>
6.1	Considerações Finais . . . . .	105
6.2	Trabalhos Futuros . . . . .	107
	<b>Referências</b>	<b>110</b>

# Lista de Figuras

2.1	Exemplos dos conceitos de grafos: (a) grafo não direcionado $G(V, E)$ ; (b) sub-grafo de $G$ ; (c) componentes biconectados de $G$ ; (d) grafo resultante após a remoção da articulação $v_6$ do grafo $G$ ; (e) grafo resultante após a remoção da ponte $v_9v_{10}$ do grafo apresentado $G$ ; e (f) árvore criada a partir da remoção de arestas do grafo $G$ . . . . .	12
2.2	Exemplo para localização de articulações utilizando Hopcroft e Tarjan [22]: (a) grafo original; e (b) árvore criada com o DFS e identificação da articulação. . . . .	13
2.3	Exemplo para localização de articulações utilizando Tarjan [54]: (a) valor do DFN atribuído a cada nó (b) valor de $L$ para cada nó e identificação da articulação. . . . .	14
2.4	Algoritmo localizado para detectar articulações: (a) grafo $G(V, E)$ , (b) executando o algoritmo em $v_4$ e $v_7$ com $k = 1$ ; e (c) executando o algoritmo em $v_4$ e $v_7$ com $k = 2$ . . . . .	18
2.5	Exemplo para o algoritmo distribuído de Chaudhuri [14] para detectar articulações: (a) topologia da rede; (b) construção da árvore; (c) propagação dos ancestrais; (d) propagação dos ciclos e definição das articulações (nós $v_2$ e $v_3$ ). . . . .	20
2.6	Reorganização da rede utilizando o mecanismo proposto por Khelifa <i>et al.</i> [25]: (a) grafo que representa a rede implantada; e (b) ativando nós inertes e movendo nós para regiões próximas a articulação. . . . .	23
2.7	Exemplo da redução do consumo de energia nas articulações utilizando agregação e dados: (a) agregação de dados a 1 salto; (b) exemplo da agregação de dados para dois pacotes, $PKT_1$ e $PKT_2$ . . . . .	25
3.1	Comunicação usando MIMO. . . . .	30
3.2	Exemplo de Comunicação Cooperativa. . . . .	31
3.3	Métodos de cooperação: (a) Aplicar e Encaminhar (ou AF) e (b) Decodificar e Encaminhar (ou DF). . . . .	32
3.4	Comunicação Cooperativa em única etapa. . . . .	32

3.5	Comunicação Cooperativa em duas etapas: (a) fase de coordenação; e (b) fase de cooperação. . . . .	33
3.6	Uso da Comunicação Cooperativa para extrapolar o raio de comunicação máximo. . . . .	35
3.7	Exemplo do <i>CoopBridges</i> (Fonte: [61]): (a) grafo inicial; (b) enlaces resultantes após aplicar a árvore geradora mínima sobre os enlaces cooperativos que ligam os clusters; (c) enlaces resultantes após a aplicação da árvore geradora mínima nos enlaces diretos; (d) grafo resultante após a combinação dos enlaces diretos e dos enlaces cooperativos. . . . .	40
3.8	Exemplo do <i>CoopSink</i> (Fonte: [36]): (a) grafo inicial com a potência máxima de transmissão dos nós; (b) arestas cooperativas; (c) grafo resultante do Controle de Topologia sobre os enlaces diretos; (d) grafo resultante após a combinação dos enlaces diretos remanescentes e dos enlaces cooperativos de menor custo. . . . .	43
4.1	Exemplo do algoritmo DAS. . . . .	51
4.2	Exemplo do algoritmo DBS. . . . .	55
4.3	Exemplo de pior caso para o algoritmo DBS. . . . .	57
4.4	Exemplo do algoritmo DDC. . . . .	60
4.5	Exemplo da localização dos elementos críticos e dos componentes. (a) grafo UDG obtido com $P_{MAX}$ ; (b) árvore distribuída; (c) articulações; e (d) articulações e pontes. . . . .	62
4.6	Quantidade de grafos conectados em função do raio de transmissão e densidade. . . . .	63
4.7	Quantidade de articulações e pontes por densidade. . . . .	64
4.8	Quantidade de articulações e pontes por densidade. . . . .	65
4.9	Quantidade de articulações e pontes para $d = 70$ em uma área de $300 \times 300m$ . . . . .	66
5.1	Utilização do CARCC2h para recuperar a conectividade para falha em: (a) articulação; e (b) ponte. . . . .	70
5.2	CARCC2h: Recuperação de conectividade por falha em articulação. . . . .	74
5.3	CARCC2h: Recuperação de conectividade por falha em ponte. . . . .	74
5.4	CARCC2h: Potência de transmissão para a fase de coordenação. . . . .	77
5.5	CARCC2h: Potência de transmissão para a fase de cooperação. . . . .	79
5.6	CARCC2h: Potência de transmissão relativa. . . . .	80
5.7	CARCC2h: Número de auxiliares. . . . .	82
5.8	CARCC2h: Potência de transmissão total. . . . .	83
5.9	CARCC2h: Potência de transmissão sem o limite $P_{MAX}$ . . . . .	84

5.10	CARCC2h: Custo computacional em relação do uso do DBHSS. . . . .	85
5.11	CARCC: Exemplo de execução do algoritmo. . . . .	91
5.12	Exemplo de simulação do CARCC: (a) grafo inicial; (b) grafo resultante após a remoção de uma articulação; (c) enlaces cooperativos criados sem otimização (linhas 1-9); e (d) enlace cooperativo criado após a seleção da origem e ajuste de potência (linhas 1-15). . . . .	94
5.13	CARCC: Potência de transmissão na fase de coordenação. . . . .	95
5.14	CARCC: Potência de transmissão na fase de cooperação. . . . .	96
5.15	CARCC: Número de auxiliares. . . . .	97
5.16	CARCC2h: Potência de transmissão relativa. . . . .	99
5.17	CARCC: Potência de transmissão total. . . . .	100
5.18	CARCC: Potência de transmissão sem o limite $P_{MAX}$ . . . . .	101
5.19	CARCC: Número de enlaces criados sem otimização (linhas 1-9). . . . .	102

# Lista de Tabelas

2.1	Mecanismos para localização de articulações em redes sem fio. . . . .	22
2.2	Volume de dados trafegados na rede. . . . .	25
2.3	Gasto médio de energia ( $mJ$ ) . . . . .	26
2.4	Mecanismos para mitigar os efeitos da presença de articulações. . . . .	26
3.1	Controle de topologia baseado em Comunicação Cooperativa. . . . .	44
5.1	CARCC2h: Relação da potência de transmissão utilizada. . . . .	86

# Lista de Abreviaturas e Siglas

**AF** *Amplify and Forward.*

**AODV** *Ad Hoc On-Demand Distance Vector.*

**CARCC** *Connectivity Auto-Recovering through Cooperative Communication.*

**CARCC2h** *Connectivity Auto-Recovering through Cooperative Communication Using 2-hop Information.*

**CC** *Comunicação Cooperativa.*

**CT** *Controle de Topologia.*

**DAS** *Distributed Articulation Search.*

**DBHSS** *Distance Based Helper Set Selection.*

**DBS** *Distributed Bridge Search.*

**DDBC** *Distributed Define Biconnected Components.*

**DDFS** *Distributed Depth First Search.*

**DF** *Decode and Forward.*

**DFN** *Depth First Number.*

**DFS** *Depth First Search.*

**DMST** *Distributed Minimum Spanning Tree.*

**DTCC** *Distributed Topology Control Using CC.*

**EDBHSS** *Estimated Distance Based Helper Set Selection.*

**ETCC** *Energy-Efficient Topology Control Problem With CC.*



**GHSS** *Greedy Helper Set Selecion.*

**GPS** *Global Positioning System.*

**ITCC** *Incremental Topology Control Using CC.*

**LMST** *Localized Minimum Spanning Tree.*

**MIMO** *Multiple-Input Multiple-Output.*

**MST** *Minimum Spanning Tree.*

**RSSF** *Redes de Sensores Sem Fio.*

**RSSI** *Received Signal Strength Indicator.*

**SNR** *Signal-To-Noise Ratio.*

**UDG** *Unit Disk Graph.*

# Capítulo 1

## Introdução

As redes *ad hoc* sem fio tornaram-se ferramentas importantes para auxiliar missões críticas e temporárias [34, 50, 58]. A aplicação destas redes é impulsionada pela existência de inúmeros dispositivos como telefones, *tablets*, computadores, sensores e atuadores. Estes dispositivos podem ser utilizados para oferecer uma rede de comunicação temporária, efetiva e de baixo custo para equipes de resposta a incidentes. Outras aplicações incluem a criação de redes de comunicação alternativas em caso de falha nos sistemas públicos e o monitoramento e alerta de desastres naturais [18, 19, 34]. Nas redes *ad hoc* sem fio, os nós podem se comunicar sem uma estrutura pré-definida e de forma auto-organizável [50]. A comunicação pode ocorrer a múltiplos saltos, em que os nós colaboram encaminhando pacotes. As Redes de Sensores Sem Fio (RSSF) são exemplos de redes *ad hoc* sem fio compostas por sensores limitados em processamento e memória sendo, usualmente, alimentados por baterias. As RSSF são utilizadas para coletar informações sobre o ambiente como temperatura, direção e intensidade do vento, volume de precipitação e vibrações. A comunicação ocorre por meio de rádios de curto alcance e os dados são enviados, em geral, a nós específicos chamados sorvedouros [58].

Considerando o contexto das aplicações críticas, como salvamento ou monitoramento de enchentes e deslizamentos, garantir a conectividade das redes *ad hoc* sem fio é determinante para o sucesso. Diversos trabalhos envidam esforços para estabelecer parâmetros de implantação que garantem a conectividade da rede [9, 37]. Aspectos sobre a não linearidade na conectividade dos nós, dependendo da região geográfica em que se encontram, também foram estudados [31, 43]. Informações sobre elementos críticos, como articulações e pontes, foram utilizadas para prever a segmentação da rede [13, 14, 20, 24, 25, 53, 59]. As articulações e pontes são, respectivamente, nós e enlaces críticos que, quando removidos, tornam as redes desconectadas [22, 54]. Como principal característica, eles constituem o único caminho entre conjuntos de nós da rede e apresentam um consumo desbalanceado de energia em redes a múltiplos saltos [59]. Como resultado, a rede tende a se tornar

desconectada de forma prematura.

Baseado na informação sobre a existência de nós e enlaces críticos, algumas alternativas foram consideradas para postergar a segmentação da rede. Goyal *et al.* [20] propuseram a utilização de mobilidade dos nós para recuperar a conectividade após falhas em nós críticos. Em caso de particionamento da rede, os nós se movem de forma coordenada para restabelecer a conectividade. Yong e Hu [59] apresentaram um modelo matemático para prever quando as articulações surgem com base em características da rede e das atividades dos nós. Esta proposta pode ser combinada com a solução de Li e Hou [32] que consideram a adição gradativa de nós até que a topologia resultante seja conectada. Afonseca *et al.* [2] utilizaram técnicas de agregação de pacotes para reduzir o consumo de energia dos pontos de articulação. O tempo de vida da rede aumenta em função da redução no número de *bytes* enviados ou recebidos pelos elementos críticos. Khelifa *et al.* [25] propuseram o uso de nós redundantes que podem ser movidos ou ativados em caso de falhas em articulações. Quando necessário, nós redundantes seriam ativados ou movidos para os locais apropriados, de modo a reforçar a conectividade. Apesar de seus benefícios, estas soluções requerem uma quantidade significativa de tempo e recurso para dispensar nós redundantes, ativá-los ou movimentá-los.

A Comunicação Cooperativa (CC) tem sido considerada como uma alternativa atracente para aumentar a conectividade de rede [21, 36, 60, 61]. A CC é uma técnica de diversidade espacial em que os nós compartilham suas antenas criando um sistema *Multiple-Input Multiple-Output* (MIMO) virtual [11, 21, 29]. O interesse é impulsionado pelo fato de que alguns dispositivos, como nós sensores, não suportam múltiplas antenas, em virtude de suas restrições de tamanho ou potência. Devido à capacidade de mitigar a atenuação dos sinais, esta técnica pode ser aplicada para aumentar o alcance dos nós. Song *et al.* [52] estudaram as propriedades de conectividade de redes esparsas e densas quando a CC é utilizada para estender o alcance da transmissão. Os nós que empregam CC foram capazes de melhorar a conectividade de rede mesmo em um ambiente esparso. Yu *et al.* [60] propuseram um esquema centralizado de controle de topologia para aumentar a conectividade de rede por meio do emprego da CC. Os enlaces cooperativos de menor custo são selecionados para interligar conjuntos isolados de nós na rede. Zhu *et al.* [62] aplicaram a mesma ideia num cenário em que se pode definir um limite para o consumo de energia da rede. Assim, múltiplos enlaces cooperativos são mantidos para garantir a existência de caminhos de menor custo entre os nós. Em um contexto semelhante, Neves *et al.* [36] propuseram o uso de enlaces cooperativos para criar caminhos de menor custo com o nó sorvedouro de uma RSSF. Os trabalhos de Yu *et al.* [60], Zhu *et al.* [62] e Neves *et al.* [36] consideram que a topologia inicial da rede é desconectada e demonstram que, se aplicada adequadamente, a conectividade da rede pode ser reforçada por meio do

uso da CC. Porém, estas técnicas utilizam informação global da topologia, necessitam de informações sobre a posição dos nós (geralmente obtidas por meio de *Global Positioning System* (GPS) ou Sistema de Posicionamento Global), apresentam um alto custo computacional e, por isso, são inviáveis para a recuperação de conectividade durante a operação da rede.

O desenvolvimento de mecanismos para garantir a conectividade por meio da colaboração dos nós é fundamental para consolidar as redes *ad hoc* e de sensores sem fio como tecnologias emergentes em atividades críticas e situações adversas. Neste sentido, este trabalho tem como objetivo apresentar alternativas para a recuperação de conectividade da rede, aplicando a Comunicação Cooperativa para criar enlaces adicionais em caso de falha dos elementos críticos. Ao contrário das soluções existentes, as propostas apresentadas neste trabalho não assumem mobilidade coordenada ou o uso de nós adicionais. Informações localizadas sobre a topologia foram utilizadas para reduzir o custo computacional e permitir sua aplicação durante a operação da rede.

A recuperação da conectividade é realizada em duas fases: a primeira consiste na localização de elementos críticos e componentes biconectados; e a segunda pelo estabelecimento de enlaces cooperativos sobressalentes entre os componentes biconectados. Componentes biconectados são os conjuntos máximos de nós (com o maior número possíveis de elementos) que não contém articulações. Para a primeira fase, o algoritmo distribuído para a localização de articulações de Chaudhuri [14] foi otimizado, resultando em uma redução de 50% no volume de mensagens. As pontes e os componentes biconectados, são determinados por meio de dois algoritmos que aplicam uma abordagem diferente da literatura. As informações sobre as articulações são aplicadas para detectar as pontes. De forma similar, as informações sobre articulações e pontes são aplicadas para determinar os componentes biconectados. Os algoritmos são executados de forma incremental, conforme a necessidade, e a interação ocorre por meio de variáveis de memória. Para a segunda fase, são apresentadas duas soluções distribuídas e localizadas capazes de oferecer resiliência à conectividade da rede de forma auto-organizável. Neste trabalho, resiliência significa que a rede é capaz de recuperar a conectividade, tornando-se tolerante a falhas, enquanto auto-organizável indica que os nós são capazes de criar enlaces sobressalentes de forma colaborativa com autonomia de decisão. A primeira solução utiliza informações a 2 saltos, a partir das articulações, para computar enlaces cooperativos entre os componentes biconectados. As articulações são monitoradas e, quando falham, o enlace cooperativo é imediatamente criado. A segunda solução não requer a coleta de informações adicionais sobre a topologia e, a partir de uma estimativa arbitrária de distância entre os componentes biconectados, é capaz de criar múltiplos enlaces cooperativos. Em seguida, o número de enlaces é reduzido por meio do ajuste na potência de transmissão. Como

principal resultado, este trabalho demonstra que é possível utilizar a CC no contexto de recuperação de conectividade por meio da redução no custo computacional. Ao aplicar o conhecimento global da topologia, o melhor enlace cooperativo, entre componentes isolados da rede, tem um custo computacional de  $O(|V|^2 \cdot \Lambda)$ , onde  $|V|$  corresponde ao número de nós da rede e  $\Lambda$  equivale ao custo para selecionar os nós auxiliares em um enlace cooperativo. A primeira solução requer  $O(\Delta(G)^2 \cdot \Lambda)$  unidades de tempo para determinar, a partir das articulações, o melhor enlace cooperativo localizado a dois saltos, onde  $\Delta(G)$  representa o grau da rede. Na segunda solução, os vizinhos das articulações computam o enlace cooperativo de reserva com um custo total de  $O(\Delta(G) \cdot \Lambda)$ . Os enlaces também podem ser aplicados para garantir a biconectividade da rede e realizar, por exemplo, o balanceamento de carga. Apesar dos enlaces cooperativos necessitarem de uma maior potência de transmissão, em relação aos enlaces diretos, os nós possuem limites para a potência de transmissão que os impedem de garantir a conectividade da rede. Os resultados demonstram que, ao cooperarem, individualmente os nós não ultrapassam 63% da potência máxima de transmissão.

No texto que segue, o problema selecionado para este trabalho é definido, são apresentados os objetivos gerais e específicos, a metodologia utilizada é detalhada e são listadas as contribuições deste trabalho no estado da arte. Por fim, a estrutura do trabalho é enumerada e detalhada.

## 1.1 Definição do Problema

As redes *ad hoc* sem fio são utilizadas em diversos cenários que exigem resiliência da rede. As técnicas da literatura para garantir a conectividade utilizam informações sobre a presença de nós e enlaces críticos como uma forma de prever a ruptura e a adoção de mobilidade coordenada ou a ativação de nós inertes para garantir a conectividade [14, 20, 24, 25, 56]. Estas soluções nem sempre são viáveis considerando o custo para disponibilizar nós extras e características especiais para oferecer mobilidade. A Comunicação Cooperativa se torna uma alternativa interessante, permitindo que os nós cooperem no formato de um sistema MIMO virtual, e obtenham um ganho no alcance de transmissão por meio do combate ao desvanecimento de sinal.

Esta tese aborda o problema da “recuperação de conectividade em rede *ad hoc* sem fio por meio da comunicação cooperativa após a falha de elementos críticos (articulações e pontes)”. Pela definição, na indisponibilidade de um elemento crítico, a rede se torna desconectada, isto é, passam a existir dois ou mais conjuntos de nós que não podem comunicar entre si.

## 1.2 Objetivos

O objetivo deste trabalho é apresentar alternativas para a recuperação de conectividade em redes *ad hoc* sem fio, empregando a Comunicação Cooperativa de forma proativa, auto-organizável e com baixo custo computacional em relação a tempo e número de mensagens. Informações sobre a presença de articulações e pontes devem ser utilizadas para prever a segmentação da rede. Os objetivos específicos compreendem:

- Propor mecanismos distribuídos e de baixo custo computacional, em termos de tempo e número de mensagens, para identificar os nós e enlaces críticos, bem como o componente biconectado de cada nó;
- Propor mecanismos de baixo custo computacional para selecionar os nós aptos a retransmitirem pacotes durante a CC;
- Elaborar mecanismos distribuídos e de baixo custo para prever a segmentação da rede, durante a operação e, recuperar a conectividade por meio da CC.

## 1.3 Metodologia

A metodologia aplicada neste trabalho será composta por uma revisão do estado da arte, pela elaboração de propostas para solucionar o problema escolhido, pela avaliação das soluções por meio de simulações e pela comparação dos resultados com propostas análogas ou relacionadas. Esta seção descreve cada uma das etapas da metodologia aplicada para permitir a replicação dos resultados.

A revisão do estado da arte terá como base, a busca, seleção e leitura crítica de artigos publicados em conferências e jornais científicos indexados pelo *Institute of Electrical and Electronics Engineers* (IEEE)<sup>1</sup>, pela *Association for Computing Machinery* (ACM)<sup>2</sup>, pela Elsevier ScienceDirect<sup>3</sup>, pela Springer<sup>4</sup> e por meio do Google Scholar<sup>5</sup>. Livros clássicos da área de redes de computadores e algoritmos também serão utilizados. A revisão bibliográfica consistirá em avaliar as premissas, os requisitos, as características e os resultados dos trabalhos relacionados aos seguintes temas: (i) definições e conceitos teóricos sobre as articulações, pontes e componentes biconectados; (ii) detecção de articulações e pontes e neutralização de seus efeitos em redes *ad hoc* sem fio; (iii) conceitos básicos de Comunicação Cooperativa (CC); e (iii) uso da CC como mecanismo para estender o raio de comunicação dos nós e garantir a conectividade da rede.

---

<sup>1</sup><http://ieeexplore.ieee.org/>.

<sup>2</sup><http://dl.acm.org/>.

<sup>3</sup><http://www.sciencedirect.com/>.

<sup>4</sup><http://link.springer.com/>.

<sup>5</sup><http://scholar.google.com/>.

As soluções serão validadas por simulações no Matlab<sup>®</sup>, por meio de um conjunto de topologias geradas de forma aleatória. Um conjunto de métricas e parâmetros de simulação serão definidos com base nos trabalhos [36, 45, 60, 61, 62] para permitir a comparação de resultados e validação das técnicas estudadas e dos mecanismos propostos. O desempenho será verificado por meio de versões análogas que realizam processamento centralizado e aplicam o conhecimento global da topologia.

## 1.4 Contribuições

Considerando os objetivos propostos e a revisão bibliográfica, a principal contribuição deste trabalho é demonstrar que é possível utilizar a Comunicação Cooperativa para manter a conectividade da rede durante a operação. Este aspecto é corroborado pela redução no custo computacional para estabelecer enlaces cooperativos entre componentes de rede. Sob o aspecto dos elementos críticos, este trabalho se destaca pela redução no número de mensagens (e conseqüentemente no volume de dados) necessárias para identificar articulações de forma distribuída, em relação ao trabalho de Chaudhuri [14]. A combinação das soluções propostas oferece resiliência à conectividade das redes *ad hoc* sem fio de forma auto-organizável, distribuída e com baixa complexidade computacional.

## 1.5 Organização do Trabalho

O restante deste trabalho é constituído por revisão do estado da arte e por soluções propostas. Para facilitar a compreensão, os Capítulos 2 e 3 agrupam a revisão bibliográfica, respectivamente, para a presença de elementos críticos em redes sem fio e o uso da Comunicação Cooperativa para incrementar da conectividade. Os Capítulos 4 e 5 apresentam, respectivamente, as soluções propostas nesta tese para a localização dos elementos críticos e para a recuperação de conectividade. Os capítulos estão organizados da seguinte forma:

- Capítulo 2: apresenta os desafios em relação à presença de articulações e pontes em redes *ad hoc* sem fio. Inicialmente, são apresentados os elementos da teoria de grafos utilizados no trabalho, incluindo definições formais para articulação e ponte. Em seguida, são apresentados diferentes mecanismos para a localização das articulações em redes sem fio. Finalmente, são detalhados alguns dos mecanismos usados para minimizar os efeitos dos elementos críticos em redes *ad hoc* sem fio.
- Capítulo 3: introduz a Comunicação Cooperativa e um modelo genérico que facilita a implementação, coleta de dados e análise de resultados. Em seguida, são apre-

sentadas as soluções da literatura para aumentar a conectividade em redes sem fio empregando enlaces cooperativos.

- Capítulo 4: apresenta uma solução distribuída para localizar articulações, pontes e componentes biconectados em uma rede *ad hoc* sem fio. A solução é analisada sob o aspecto do custo computacional. Dados numéricos são utilizados para definir os parâmetros de simulação durante a recuperação de conectividade.
- Capítulo 5: apresenta duas soluções capazes de recuperar a conectividade, após a falha de elementos críticos, por meio da Comunicação Cooperativa de forma auto-organizável e usando informações localizadas da topologia.
- Capítulo 6: resume os resultados obtidos, apresenta uma conclusão sobre o trabalho e discute possibilidades para trabalhos futuros.



## Capítulo 2

# Articulações e Pontes em Redes *Ad Hoc* Sem Fio

As articulações e pontes são conceitos amplamente discutidos na teoria de grafos. Informalmente, uma articulação é um nó cuja remoção (consequentemente também de suas arestas) torna o grafo desconectado. De forma análoga, uma ponte é uma aresta cuja remoção torna o grafo desconectado. Devido à importância na conectividade, as articulações e pontes são também conhecidas, respectivamente, como nós críticos e enlaces críticos [53]. Nas redes sem fio, as articulações e pontes têm se destacado como foco de estudo, considerando que suas propriedades têm impactos em questões como eficiência e tempo de vida da rede [59]. Estes elementos também estão relacionados ao conceito de biconectividade em redes de comunicação, cuja importância se destaca em função da resiliência e do balanceamento de carga [46]. Tendo em vista sua relevância, vários trabalhos apresentam soluções para sua localização e para mitigar os efeitos causados pela sua presença nas redes sem fio.

Neste trabalho, os elementos críticos são utilizados para prever a segmentação da rede e determinar onde os enlaces devem ser criados por meio da Comunicação Cooperativa. Assim, este capítulo apresenta uma revisão teórica sobre elementos críticos e trabalhos relacionados no contexto das redes sem fio. A Seção 2.1 apresenta as noções da teoria de grafo necessárias para a definição formal das articulações e pontes, assim como algoritmos clássicos para detectar estes elementos. Em seguida, na Seção 2.2, são apresentados os mecanismos da literatura para localizar as articulações em redes *ad hoc* sem fio. Na Seção 2.3 são discutidas as soluções da literatura que tem como objetivo mitigar os problemas causados pela presença dos elementos críticos, em especial a recuperação de conectividade. A Seção 2.4 resume este capítulo e apresenta uma discussão sobre os desafios relacionados a área.

## 2.1 Articulações e Pontes

Grafos são um conjunto de vértices (pontos ou nós) e outro de arestas que conectam pares de vértices [51]. Um grafo pode ser representado por  $G(V, E)$  onde  $V$  é o conjunto de vértices e  $E$  é o conjunto de arestas, tal que  $(v_i v_j) \in E$  é uma aresta que interliga os vértices  $v_i$  e  $v_j$ ,  $v_i, v_j \in V$ . A quantidade de vértices pode ser representada por  $|V|$ . De forma análoga, a quantidade de arestas é representada por  $|E|$ . Para cada aresta  $(v_i v_j)$  pode-se atribuir um peso, ou custo, utilizando uma função  $w(v_i v_j)$ . Dado a definição de grafos, diversos conceitos relacionados, extensivamente aplicados neste trabalho, são apresentados a seguir.

Um grafo em que se indica uma direção para cada aresta é chamado de grafo direcionado ou orientado. Quando a direção das arestas não é indicada, denomina-se grafo não direcionado. Em um grafo não direcionado, o grau de um nó  $v_i$ , ou  $\Delta(v_i)$ , corresponde ao número de arestas em que o nó  $v_i$  participa (ou incidentes em  $v_i$ ) [39]. O grau de um grafo não direcionado corresponde ao maior valor de grau dos seus nós, e pode ser determinado pela equação:

$$\Delta(G) = \max(\Delta(v_i)), v_i \in V. \quad (2.1)$$

Dizemos que dois nós são conectados se existe um caminho entre eles. Um caminho é uma sequência de arestas entre dois nós  $v_i$  e  $v_j$  na forma  $v_i v_{i+1}, v_{i+1} v_{i+2}, \dots, v_{j-1} v_j$  tal que todas as arestas e todos os vértices intermediários são diferentes [8]. Pode-se definir um peso para o caminho pela soma de todas as arestas que o compõem. Um ciclo é uma sequência de arestas que inicia e termina no mesmo vértice na forma  $v_i v_{i+1}, v_{i+1}, v_{i+2}, \dots, v_j v_i$  tal que todas as arestas e todos os vértices intermediários são diferentes [8]. Em um grafo não direcional com  $|V| > 3$ , a presença de ciclos implica na existência de múltiplos caminhos entre pares de nós [35]. Dado dois vértices quaisquer, os caminhos entre eles são comparados pelos custo.

Um grafo  $G(V, E)$  é conectado se existe um caminho entre cada par de vértices  $v_i, v_j \in V$ ,  $v_i \neq v_j$  [39]. De forma análoga, um grafo  $G(V, E)$  é desconectado se não existe um caminho entre cada par de vértices. Um grafo  $G'(V', E')$  é um subgrafo de  $G(V, E)$ , se  $V' \subset V$  e  $E' \subset E$  [17]. Os vértices que podem ser alcançados por  $v_i$  bem como as arestas incidentes a eles são chamados de componente conectado de  $v_i$  [39]. Assim, componentes conectados são conjuntos disjuntos de elementos, ou subgrafos sem conectividade entre si. A Definição 2.1.1 apresenta formalmente o conceito de componentes conectados.

**Definição 2.1.1.** *(Componente conectado [51]) Um grafo não conectado consiste em um conjunto de componentes conectados que são subgrafos conectados máximos (com o maior número possível de elementos).*

Em outras palavras, um componente conectado  $G_a(V_a, E_a)$ , ou simplesmente componente, de um grafo  $G(V, E)$  não direcional é um subgrafo tal que qualquer vértice  $v_i \in V_a$  está conectado a todos os outros vértices  $v_j \in V_a$  por meio de um caminho e que, dado dois componentes diferentes  $G_a(V_a, E_a)$  e  $G_b(V_b, E_b)$ , eles não possuem vértices em comum, isto é,  $V_a \cap V_b = \emptyset$ . Um grafo conectado possui um único componente. Em um grafo desconectado existem mais de um componente, cuja quantidade varia de 2 a  $|V|$ . Note que um subgrafo não é, necessariamente, um componente conectado, porém, um componente conectado é um subgrafo. Um grafo não orientado é dito fortemente conectado quando existe um caminho entre cada par de vértices. Um grafo é dito  $k$ -conectado, quando  $k$  é o menor número de vértices cuja remoção torna o grafo desconectado [8].

Uma árvore é um grafo conectado que não possui ciclos. Em particular, existe um único caminho conectado entre qualquer par de vértices [39]. Componentes conectados de grafos sem ciclos são árvores. Uma árvore geradora  $T(V, E')$  é um subgrafo conectado de  $G(V, E)$ , tal que inclui todos os vértices e não existem ciclos. Uma *Minimum Spanning Tree* (MST), ou árvore geradora mínima, é aquela em que a soma de todos os pesos das arestas não é maior que qualquer outra árvore geradora.

Pode-se transformar um grafo em árvore eliminando arestas que fazem parte de ciclos. Como propriedade, dado uma árvore qualquer com  $n$  vértices, existem  $(n - 1)$  arestas [8]. Assim, dado um grafo  $G$  com  $n$  vértices e  $m$  arestas, então uma árvore  $T$  criada a partir da remoção dos ciclos de  $G$  tem o mesmo número de vértices do grafo original  $G$  e, como possui  $(n - 1)$  arestas, então  $\gamma = m - n + 1$  arestas foram removidas. Neste trabalho, assim como em [14], as arestas removidas durante a geração da árvore  $T$  a partir de  $G$  são chamadas de arestas “*nontree*” e são representadas como  $\gamma(G)$ . Os algoritmos de Prim e Kruskal são exemplos de técnicas para computar MST [17].

Sedgewick *et al.* [51] e Cormen *et al.* [17] definem articulação como um vértice  $v \in V$  de um grafo conectado  $G(V, E)$  cuja remoção (e conseqüentemente de suas arestas) torna o grafo  $G$  desconectado. De forma análoga, uma ponte é uma aresta  $(v_i v_j) \in E$  tal que sua exclusão torna o grafo desconectado [17]. Note que, quando  $|V| \geq 3$ , a existência de uma ponte implica na existência de mais de uma articulação no grafo. Os conceitos de articulação e ponte são apresentados formalmente nas Definições 2.1.2 e 2.1.3.

**Definição 2.1.2.** (*Articulação [35]*) Dado um grafo conectado  $G(V, E)$ , o nó  $v_a$  é uma articulação se  $G'(V', E')$  é um grafo desconectado para  $V' = V \setminus \{v_a\}$  e  $E' = E \setminus \{(v_a v_i)\}$  para todo  $v_i \in V$ .

**Definição 2.1.3.** (*Ponte [35]*) Dado um grafo conectado  $G(V, E)$ , a aresta  $v_a v_b$  é uma ponte se  $G'(V, E')$  é um grafo desconectado para  $E' = E \setminus \{(v_a v_b)\}$ .

Um grafo que possui articulações é chamado de grafo separável, enquanto um grafo sem articulações é dito grafo não separável ou biconectado [46]. Em [47] e [54], são

apresentadas diversas propriedades sobre as articulações, em especial, demonstra-se que um grafo sem articulações é biconectado. Um componente biconectado, ou “bloco”, de  $G$  é um conjunto máximo de arestas tal que qualquer par de arestas no conjunto estão em um mesmo ciclo [17]. Um componente biconectado também pode ser definido em função de articulações, conforme a Definição 2.1.4.

**Definição 2.1.4.** (*Componente Biconectado [35]*) Um componente  $G_a(V_a, E_a)$  é dito componente biconectado, ou bloco, quando é o maior conjunto (ou subgrafo máximo), e não possui articulações.

Note que um grafo conectado que possui apenas dois nós é, pela definição, um grafo biconectado. De forma semelhante, um subgrafo conectado com dois nós é um componente biconectado. Em [35] denota-se que um grafo ou componente é 2-conectado se existe um ciclo entre qualquer par de nó. A Definição 2.1.5 apresenta, formalmente, o conceito de componente 2-conectado. Neste caso, um grafo ou componente conectado composto em que  $|V| = 2$  é um componente biconectado, mas não é 2-conectado. Um grafo ou componente biconectado em que  $|V| > 2$  é também 2-conectado.

**Definição 2.1.5.** (*Componente 2-conectado [35]*) Um componente  $G_a(V_a, E_a)$  de  $G(V, E)$ , tal que  $V_a \subset V$  e  $E_a \subset E$ , é 2-conectado se, e somente se, para qualquer par de vértices  $v_i, v_j \in V_a$ , existe um ciclo em  $E_a$  contendo  $v_i$  e  $v_j$ .

A Figura 2.1 ilustra os conceitos apresentados. Na Figura 2.1(a), o grafo  $G(V, E)$  apresentado possui  $|V| = 14$  vértices e  $|E| = 20$  arestas. Neste grafo, os nós  $v_6, v_9$  e  $v_{10}$  são articulações. A aresta  $v_9v_{10}$  é uma ponte. A Figura 2.1(b) ilustra um subgrafo  $G'(V', E')$  a partir do grafo apresentado na Figura 2.1(a). Este subgrafo possui  $|V'| = 5$  vértices e  $|E'| = 5$  arestas. Note que  $V' \subset V$  e  $E' \subset E$ . Este subgrafo também é um componente conectado e pode ser denotado por  $G_a(V_a, E_a)$ . Como este componente possui articulações, não é biconectado. Na Figura 2.1(c), são representados os componentes biconectados e 2-conectados do grafo apresentado em 2.1(a). Neste exemplo, existem quatro componentes biconectados e três componentes 2-conectados. A Figura 2.1(d) apresenta o grafo resultante da remoção de articulação  $v_6$ . Note que, agora, o grafo possui dois componentes conectados. Os conjuntos  $\{v_7, v_9\}$  e  $\{v_8, v_9\}$  constituem novos componentes biconectados e, portanto, o grafo resultante possui cinco componentes biconectados e somente dois componentes 2-conectados. Na Figura 2.1(e) é representado o grafo resultante da remoção da ponte  $v_9v_{10}$  do grafo original, representado na Figura 2.1(a). O grafo resultante é composto por dois componentes conectados e três componentes biconectados/2-conectados. A Figura 2.1(f) representa uma árvore criada a partir da remoção de  $\gamma = 7$  arestas do grafo original apresentado em 2.1(a). As linhas pontilhadas representam as arestas “*nontree*”.

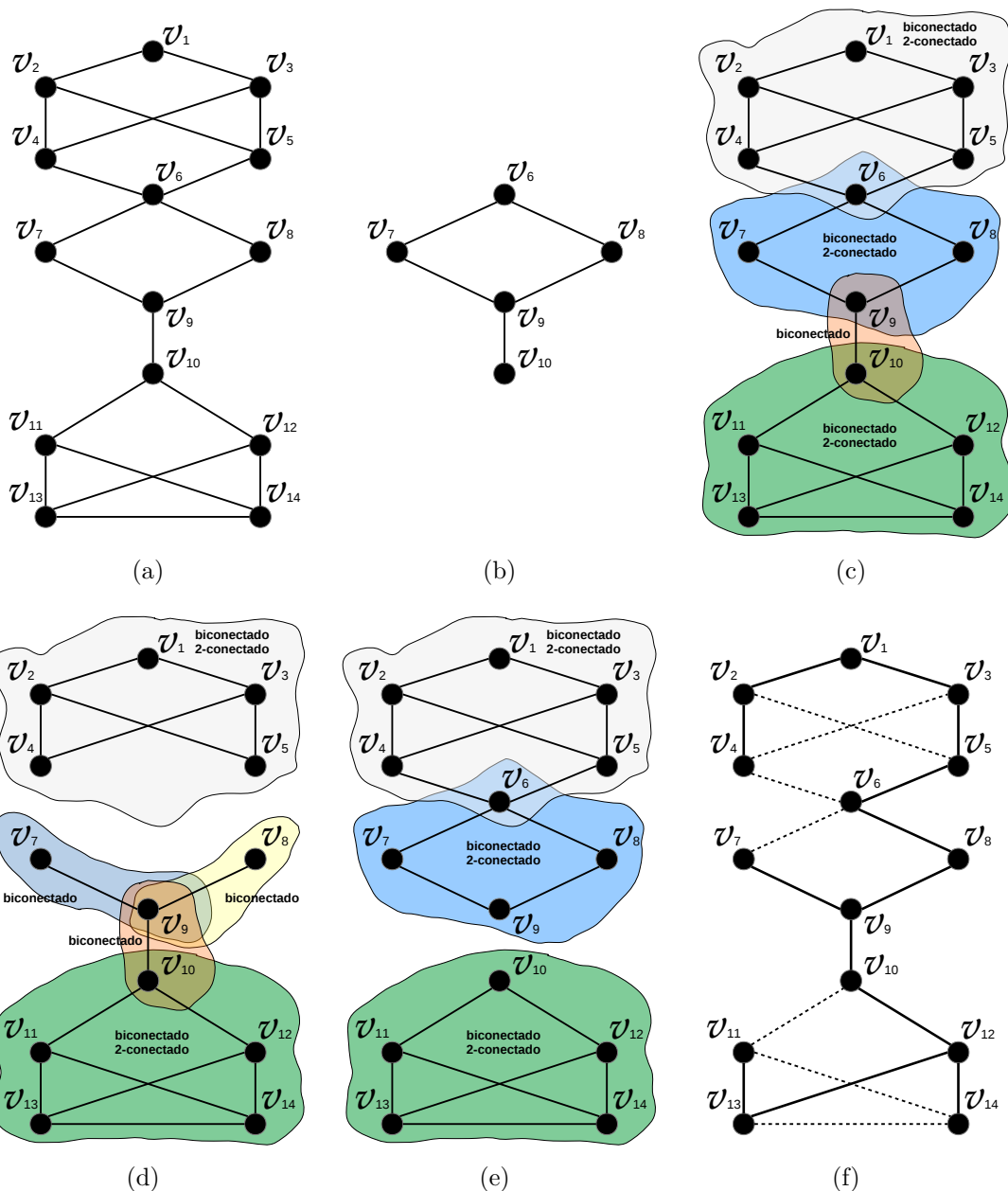


Figura 2.1: Exemplos dos conceitos de grafos: (a) grafo não direcionado  $G(V, E)$ ; (b) sub-grafo de  $G$ ; (c) componentes biconectados de  $G$ ; (d) grafo resultante após a remoção da articulação  $v_6$  do grafo  $G$ ; (e) grafo resultante após a remoção da ponte  $v_9v_{10}$  do grafo apresentado  $G$ ; e (f) árvore criada a partir da remoção de arestas do grafo  $G$ .

Um algoritmo ingênuo para localizar articulações em um grafo  $G(V, E)$  conectado consiste em, para cada nó  $v_i \in V$ , removê-lo e verificar se o grafo resultante é conectado. Remover cada nó do grafo tem um custo de  $O(|V|)$ . Verificar a conectividade de um grafo pode ser realizada em  $O(|V| + |E|)$  [51]. Assim, o algoritmo ingênuo não é eficiente, já que sua complexidade é de  $O(|V| * (|V| + |E|))$ . Hopcroft e Tarjan [22] e Tarjan [54] desenvolveram algoritmos com complexidade linear que aplicam árvores geradoras para

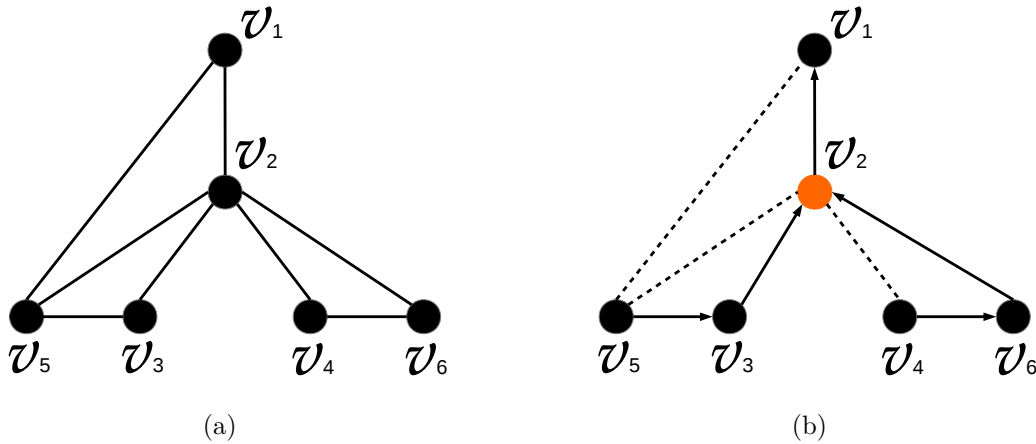


Figura 2.2: Exemplo para localização de articulações utilizando Hopcroft e Tarjan [22]: (a) grafo original; e (b) árvore criada com o DFS e identificação da articulação.

detectar as articulações com tempo  $O(|V| + |E|)$ . Em [22], uma árvore é produzida a partir do grafo original, utilizando o algoritmo *Depth First Search* (DFS) ou Algoritmo de Busca em Profundidade. Nesta metodologia, as articulações são nós que não possuem, para cada filho, uma aresta que liga um descendente (naquele ramo) a um ancestral. Como caso específico, se a raiz possui mais de um filho, é também uma articulação. Este mecanismo é baseado em um conjunto de lemas cujas provas são apresentadas em [14] e no trabalho original [22]. A Figura 2.2 exemplifica a localização de articulações utilizando DFS e as premissas apresentadas em [22]. A Figura 2.2(a) apresenta o grafo  $G(V, E)$  inicial com  $|V| = 6$  nós e  $|E| = 8$  arestas. A Figura 2.2(b) mostra a árvore  $T(V, E')$  criada a partir do grafo  $G(V, E)$  da Figura 2.2(a). Note que  $E' = E \setminus \{v_1v_5, v_2v_5, v_2v_4\}$ . As arestas removidas, ou arestas *nontree*, estão representadas por meio de linhas pontilhadas. As setas, nas arestas da árvore, representam a hierarquia e sua direção determina o nó pai. O nó  $v_2$ , em destaque, é uma articulação já que o nó filho  $v_6$  e seus descendentes não possuem uma aresta para um ancestral de  $v_2$ .

Tarjan [54] elaborou uma técnica similar, baseada no *Depth First Number* (DFN), ou Ordem na Busca em Profundidade, para determinar as articulações de um grafo. Ao realizar a busca em profundidade, cada nó  $v_i$  é marcado com um número  $DFN(v_i)$  que corresponde a ordem em que foi adicionado à árvore. Em seguida, a cada nó é atribuído um valor  $L(v_i)$  indicando o menor valor de DFN que pode ser atingido por aquele nó por meio de um caminho que podem incluir um número arbitrário de arestas pertencentes ou não à árvore. Um nó  $v_i$  é marcado como articulação quando um de seus filhos  $v_j$  possuírem um valor  $L(v_j)$  maior ou igual a  $DFN(v_i)$ , ou se este nó for a raiz e possuir mais de um filho. A Figura 2.3 exemplifica a técnica proposta por Tarjan [54]. A Figura 2.3(a) ilustra a árvore criada a partir do grafo  $G(V, E)$  da Figura 2.2(a) e, ao lado de cada

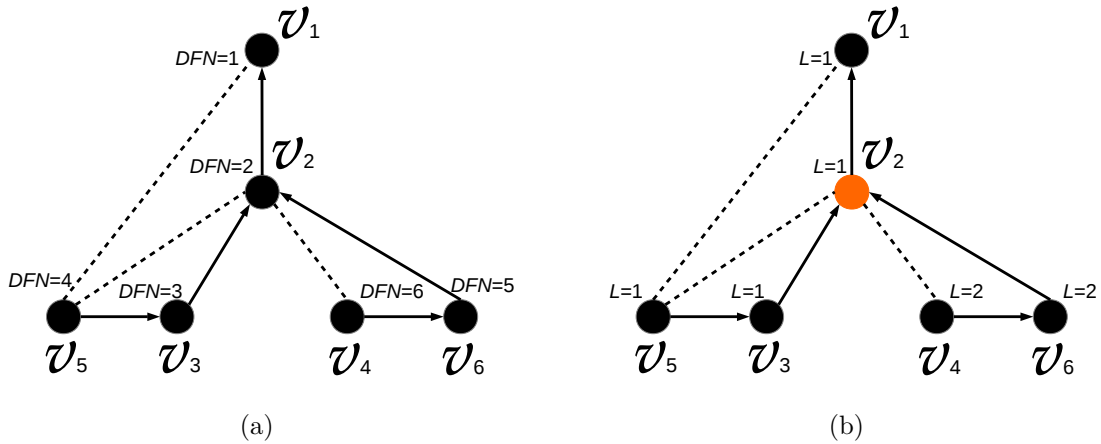


Figura 2.3: Exemplo para localização de articulações utilizando Tarjan [54]: (a) valor do DFN atribuído a cada nó (b) valor de  $L$  para cada nó e identificação da articulação.

nó, um número que representa a ordem em que o nó foi inserido na árvore. Na Figura 2.3(b), é identificado o valor de  $L$  para cada nó. Neste caso, o nó  $v_6$  tem um número  $L = 2$  indicando que este é o menor valor  $DNF$  encontrado por ele e seus descendentes por meio de caminhos partindo de suas arestas. O nó  $v_2$ , em destaque, é uma articulação já que, conforme o algoritmo de Tarjan [54],  $L(v_6) = 2 \geq DNF(v_2) = 2$ .

## 2.2 Localização de Articulações e Pontes em Redes *Ad Hoc* Sem Fio

Considere um grafo  $G(V, E)$  conectado tal que  $v_a \in V$  é uma articulação. O nó  $v_a$  liga dois componentes biconectados,  $G_1(V_1, E_1)$  e  $G_2(V_2, E_2)$ . Pelas definições e pelos exemplos relatados na Seção 2.1, dado um nó  $v_i \in V_1$  e um nó  $v_j \in V_2$ , qualquer caminho entre eles contém  $v_a$ . Além disso, não é possível encontrar um caminho que não inclua  $v_a$ . Em outras palavras, qualquer caminho entre um nó de um componente deve passar por  $v_a$  para chegar a um nó em outro componente. A partir deste exemplo, podemos presumir que as articulações são determinantes para a capacidade e tempo de vida das redes sem fio, já que se tornam gargalos na comunicação e apresentam um alto consumo de energia em função da retransmissão de pacotes. A localização das articulações, sobretudo após a implantação, é uma tarefa importante e que permite ações para a mitigação dos problemas causados por sua existência. O resto desta seção apresenta um modelo de comunicação simples para redes sem fio, argumenta sobre a importância da presença de articulações e apresenta mecanismos da literatura para localizar as articulações nestas redes.

Semelhante a [61], neste trabalho se utiliza um modelo simples para representar a propagação do sinal em redes sem fio por meio de dados experimentais. No modelo

genérico apresentado em [21], o sinal de rádio frequência recebido por um nó é função da potência de transmissão e da distância da origem e sofre atenuação por um fator exponencial. Para decodificar os dados, o sinal recebido pelo nó de destino deve ser superior a um determinado limiar. Assim, considere que a rede é modelada em um grafo  $G(V, E)$ , onde  $V$  representa os nós da rede e  $E$  os enlaces. Cada nó  $v_i$  pode ajustar sua potência de transmissão  $P_i$  com valores no intervalo  $[0, P_{MAX}]$ . Quando  $P_i = 0$ , o transceptor está desligado e quando  $P_i = P_{MAX}$ , o transceptor opera na potência máxima. A origem  $v_i$  pode comunicar diretamente com o nó de destino  $v_j$  somente quando a potência  $P^r$  recebida por  $v_j$ , a partir da transmissão  $P_i$  de  $v_i$ , atende a Equação 2.2:

$$P^r = P_i(d_{i,j})^{-\alpha} \geq \tau \quad (0 \leq P_i \leq P_{MAX}), \quad (2.2)$$

onde:  $\alpha$  é o expoente de desvanecimento do sinal, usualmente entre 2 e 6, que representa a taxa de perda de energia do sinal com o aumento da distância;  $d_{i,j}$  é a distância Euclidiana entre  $v_i$  e  $v_j$ ; e  $\tau$  é o valor mínimo para o *Signal-To-Noise Ratio* (SNR) (Relação Sinal-Ruído), recebido por  $v_j$ , para que o nó de destino possa decodificar o sinal recebido e obter a mensagem original.

Considere, a partir da Equação 2.2, que os nós transmitem utilizando  $P_{MAX}$  obtendo o raio de comunicação máximo  $R_{MAX}$ . Um enlace é criado quando ambos os nós estão dentro do raio de comunicação do outro. Neste caso, o grafo  $G(V, E)$  resultante corresponde ao *Unit Disk Graph* (UDG) [16], ou Grafo de Disco Unitário, em que os enlaces entre pares de nós são bidirecionais.

Nas redes *ad hoc* sem fio, a comunicação pode ocorrer a múltiplos saltos e os nós colaboram na comunicação, ao encaminhar pacotes. Nas Redes de Sensores Sem Fio (RSSF), este comportamento se destaca pela necessidade de encaminhar os dados a nós específicos chamados sorvedouros [58]. Nestes cenários, as articulações se tornam gargalos nas comunicações e um dos principais pontos de falha da rede [46]. O encaminhamento excessivo de pacotes nas articulações tem como resultado um consumo desbalanceado de energia, levando à segmentação da rede [56]. Assim, determinar as articulações, torna-se importante para prolongar a vida da rede, criar mecanismos para evitar a contenção e garantir o sucesso da aplicação.

As articulações podem surgir no processo de implantação ou durante a operação. Na implantação, parâmetros como densidade e a área de abrangência estão diretamente relacionados à conectividade e à presença de articulações. Em [26] os autores apresentam uma revisão sobre técnicas para planejar a implantação da rede e garantir a conectividade. Durante a operação, as articulações podem surgir em função de alterações na topologia como a mobilidade ou pela indisponibilidade de nós, em geral, pelo desvanecimento da bateria. Em [59], os autores modelam o surgimento de articulações em redes sem fio



durante a operação. São considerados três fatores: *(i)* fluxo: reflete a posição do nó na rede e mensura a quantidade de dados encaminhados; *(ii)* convergência: determina a importância do nó como intermediário na comunicação entre os nós e; *(iii)* surgimento de pontes: representa o grau dos nós conectados com os vizinhos de tal forma a refletir as características de convergência destes nós. Os autores apresentam uma definição formal e um modelo matemático para o surgimento de articulações. Argumenta-se que, a identificação dos nós que tendem a se tornar articulações, permite solucionar o problema do consumo desbalanceado de energia.

Considerando o modelo de comunicação da Equação 2.2, o ajuste na potência de transmissão pode ser aplicado, por exemplo, no Controle de Topologia (CT) que consiste em ajustar a potência de transmissão dos nós para reduzir a quantidade de enlaces e, conseqüentemente, minimizar a contenção, melhorar o desempenho e aumentar o tempo de vida da rede [50]. Porém, esta é uma técnica que pode acarretar o surgimento de inúmeras articulações e pontes em uma rede. Ramanathan e Rosales-Hain [44] apresentaram uma solução que permite, após aplicar o CT, aumentar gradativamente a conectividade da rede com o objetivo de reduzir o número resultante de articulações.

No contexto das redes *ad hoc* sem fio e das RSSF, vários trabalhos propõem mecanismos para a localização das articulações utilizando como base os algoritmos de Hopcroft e Tarjan [22] e Tarjan [54]. Eles podem ser classificados quanto à abordagem utilizada na representação do grafo e em relação ao processamento da seguinte forma:

- Processamento centralizado com conhecimento global: um único nó realiza o processamento sobre a representação da rede em sua memória; os algoritmos requerem a coleta de informações sobre todos os nós e enlaces da rede, os dados são utilizados para fazer uma representação única, em memória, de toda a topologia da rede. Os trabalhos [20, 25] são exemplos de algoritmos para localizar articulações através do conhecimento global e processamento centralizado.
- Processamento distribuído com conhecimento local: cada nó executa o mesmo algoritmo utilizando um conhecimento parcial da topologia; as informações coletadas da topologia (a partir de um determinado nó) são chamadas de informações a  $k$ -saltos e correspondem aos nós e enlaces conectados a um caminho com comprimento de, no máximo,  $k$  enlaces. O trabalho [24] utiliza conhecimento localizado a  $k > 1$  saltos para localizar as articulações. Quando  $k = 1$ , o nó que realiza o processamento conhece apenas os vizinhos diretamente conectados e, portanto, não é necessário obter informações adicionais sobre a topologia da rede. As soluções [14, 56] são exemplos de algoritmos localizados com  $k = 1$  e são designados, nesta tese, como “completamente distribuídos”

Goyal e Caffery [20] e Khelifa *et al.* [25] apresentam soluções centralizadas para a localização das articulações. As informações da topologia são coletadas e enviadas para um nó que reconstrói o conhecimento global da topologia e é responsável por executar o algoritmo e localizar as articulações. Em [20] os autores utilizam uma versão modificada do DFS para determinar as articulações. O método verifica, ao criar a árvore, se existe um caminho de volta em direção à raiz. Quando este caminho não existe, o nó é marcado como uma articulação. Em [25], a proposta utiliza um algoritmo centralizado que realiza uma busca em profundidade no grafo  $G(V, E)$  tal que as arestas *nontree* ( $E \setminus E'$ ), ou  $\gamma(G)$ , são utilizadas de forma semelhante ao algoritmo de Hopcroft e Tarjan [22] para verificar se o nó é uma articulação. As informações sobre a presença de articulações são utilizadas como ferramenta para prever falhas na rede e realizar a reorganização dos nós. A localização das articulações tem complexidade  $O(|V| + |E|)$ , enquanto a coleta de informação da topologia requer um tempo quadrático em função do número de nós [24]. O Algoritmo 1 enumera os passos da solução centralizada, semelhante à utilizada em [20] e [25], para localizar as articulações de uma rede sem fio.

---

**Algorithm 1** Localização Centralizada e Global de Articulações

---

- 1: Obter informação topológica da rede;
  - 2: Representar a rede utilizando um grafo  $G(V, E)$ ;
  - 3: Calcular, utilizando busca em profundidade, uma árvore  $T(V, E')$ , a partir do grafo  $G(V, E)$ , tal que  $E' \subset E$ , tomando como raiz um nó arbitrário  $v_r$ ;
  - 4: O vértice  $v_i \in V$  é uma articulação se, para qualquer nó  $v_j$  filho de  $v_i$ , não existe uma aresta que liga um descendente de  $v_j$  a um ancestral de  $v_i$ ;
  - 5: O nó raiz  $v_r$  é uma articulação se possui mais de um filho.
  - 6: Reorganizar a rede para reduzir o número de articulações;
- 

Em cenários como as RSSF, onde os nós possuem capacidade limitada, os métodos centralizados se tornam inviáveis já que necessitam de uma grande quantidade de mensagens para coletar as informações sobre a topologia da rede e que um dos nós tenha capacidade de processamento e energia para computar as articulações. Jorgic *et al.* [24], informa que estas soluções centralizadas envolvem uma quantidade quadrática de mensagens (em número de nós) para atualizar as informações de enlaces quando ocorrem mudanças na topologia. Uma alternativa é a utilização de algoritmos distribuídos que executam em cada nó e utilizam informações da topologia limitada a uma determinada quantidade de saltos em relação aos seus vizinhos. Mensagens especiais são utilizadas para coletar as informações a  $k$ -saltos. Para calcular os elementos críticos, o conhecimento coletado é aplicado em algoritmos semelhantes às soluções centralizadas. Assim como os algoritmos gulosos, estas alternativas tentam atingir o objetivo global utilizando comportamento local [24, 53].

Jorgic *et al.* [24] assumem que os nós coletam informações a  $k$ -saltos utilizando uma mensagem especial chamada “HELLO”. As informações da topologia a  $k$ -saltos são propagadas para os vizinhos a  $(k - 1)$ -saltos permitindo que os nós tenham conhecimento de parte da topologia da rede. Segundo os autores, a informação topológica a  $k$ -saltos corresponde ao subgrafo de  $k$ -vizinhos, incluindo todos os enlaces existentes entre os vizinhos a  $k$ -saltos e  $(k - 1)$ -saltos. Este conjunto de informações não inclui, estritamente, se os vizinhos  $k$ -saltos estão conectados. No método proposto pelos autores, um nó é  $k$ -crítico (ou articulação usando informações a  $k$ -saltos) se o subgrafo dos vizinhos a  $k$ -saltos é desconectado ao considerar a remoção do próprio nó. A solução é executada em cada nó da rede. Os nós devem, individualmente, coletar as informações topológicas e executar o algoritmo para determinar, por meio da representação parcial, se são articulações. Esta solução pode utilizar informações posicionais, onde a topologia da rede é determinada pela posição dos nós e o raio de transmissão. Os autores também propõem a verificação de arestas críticas a  $k$ -saltos baseado em informações sobre as articulações. A Figura 2.4 apresenta um exemplo da utilização do conhecimento localizado, nos nós  $v_4$  e  $v_7$  para localizar articulações. Na Figura 2.4(b), os nós  $v_4$  e  $v_7$  determinam que são articulações aplicando informação correspondente a  $k = 1$ . As áreas demarcadas representam o conhecimento obtido por cada nó para realizar os cálculos. Note que, conforme o exemplo, todos os nós são articulações para conhecimento a 1-salto. Na Figura 2.4(c), ao utilizar informação topológica a 2-saltos, o nó  $v_4$  determina que é uma articulação, enquanto  $v_7$  determina que é um nó ordinário. As áreas em destaque representam o conhecimento localizado dos nós  $v_4$  e  $v_7$ . Para  $v_4$ , não é possível encontrar um ciclo (múltiplos caminhos) por meio do conhecimento localizado e, por isso, o nó determina que é uma articulação.

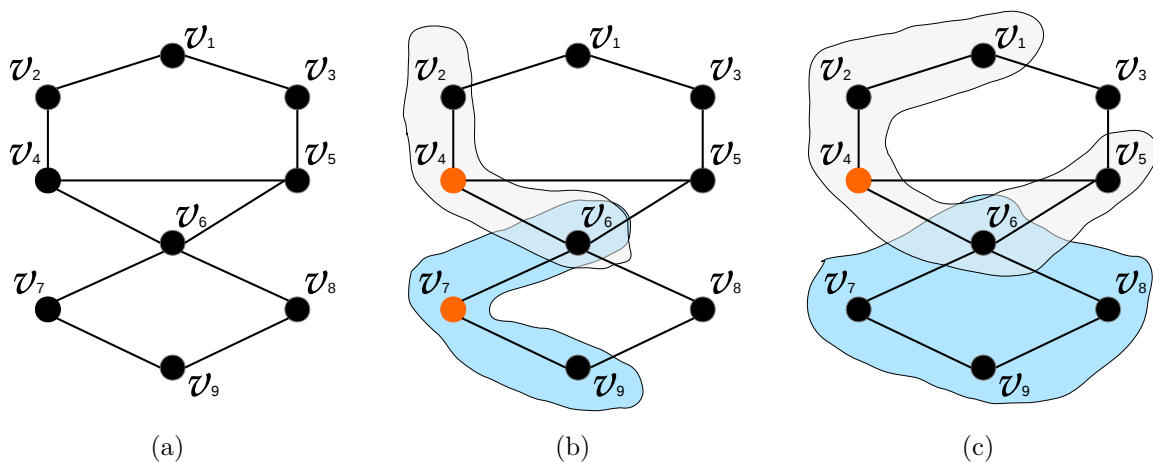


Figura 2.4: Algoritmo localizado para detectar articulações: (a) grafo  $G(V, E)$ , (b) executando o algoritmo em  $v_4$  e  $v_7$  com  $k = 1$ ; e (c) executando o algoritmo em  $v_4$  e  $v_7$  com  $k = 2$ .

Neste caso, observe que  $v_4$  produz um falso positivo já que, de fato, não é uma articulação.

Os algoritmos localizados permitem a descentralização do processamento e representação parcial do grafo. Apesar desta metodologia não reduzir a complexidade do algoritmo, os requisitos de processamento e armazenamento são reduzidos junto com o tamanho da entrada (quantidade de vértices e arestas do grafo). Em contrapartida, falsos positivos podem ser detectados. Os algoritmos completamente distribuídos utilizam, como conhecimento, apenas os enlaces incidentes e os nós diretamente conectados. Em geral, estes algoritmos localizam as articulações utilizando uma versão distribuída da busca em profundidade que representa, parcialmente, na memória dos nós, a árvore produzida pelo algoritmo e as informações sobre os ciclos existentes no grafo.

Chaudhuri [13] apresentou uma solução distribuída para determinar as pontes e os componentes biconectados de um grafo. Em seguida, o mesmo autor apresentou uma solução similar para determinar as articulações do grafo, ótimo em tempo e quantidade de mensagens [14]. Os nós computam se são articulações verificando as mesmas premissas do algoritmo proposto por Hopcroft e Tarjan [22]. O algoritmo requer que os nós tenham conhecimento apenas dos vizinhos diretamente conectados e as articulações são computadas com  $O(|V|)$  mensagens e  $O(|V|)$  unidades de tempo. Dado um grafo conectado e um nó arbitrário como raiz, esta proposta calcula as articulações em duas fases. A primeira fase consiste em construir uma árvore mantendo na memória de cada nó a definição do nó pai e dos nós filhos. Em seguida, uma lista de nós ancestrais é propagada para cada nó. Na segunda fase, a lista de arestas *nontree* é propagada das folhas para a raiz e as articulações são calculadas. A árvore é criada por meio de uma mensagem especial chamada *SEARCH*. Esta mensagem é propagada da raiz para as folhas e carrega uma lista dos nós previamente visitados. Na troca de mensagens, o destino é adicionado como nó filho pela origem, enquanto o destinatário define que a origem é o nó pai. Para determinar que um segmento foi completamente visitado, a partir de um nó, uma mensagem *SEARCH* é enviada de volta, por ele, para o nó pai. Inicialmente, esta mensagem de retorno é enviada pelas folhas da árvore que identificam, por meio da lista de nós visitados, que não existem mais nós a visitar naquele ramo. Após a criação da árvore, a raiz envia uma mensagem especial chamada *TERMINATE* para cada filho. Esta mensagem carrega a lista de nós ancestrais que é atualizada e propagada para o resto da árvore. Na segunda fase, uma mensagem *NONTREE* é enviada das folhas para os ancestrais propagando a lista de arestas que não fazem parte da árvore (arestas *nontree*). Esta solução utiliza, no máximo,  $4 \cdot |V|$  mensagens.

A Figura 2.5 apresenta um exemplo de execução da solução distribuída proposta por Chaudhuri [14]. A Figura 2.5(a) representa a topologia da rede sem fio. A Figura 2.5(b) apresenta a construção da árvore utilizando o nó  $v_1$  como raiz. Os nós são visitados se-

quencialmente por meio da mensagem *SEARCH*. A ordem em que os nós são incluídos na árvore é  $v_1, v_2, v_4, v_3, v_5$  e  $v_6$ . Note que, como o processo é sequencial, a mensagem *SEARCH* só é enviada, por exemplo, para o nó  $v_3$  quando  $v_4$  envia uma mensagem *SEARCH* informando que a busca já foi completada naquele ramo. O nó  $v_4$  determina que não existem mais nós a pesquisar utilizando o conteúdo da mensagem *SEARCH* enviada por  $v_2$ . Nesta figura também é ilustrada a estrutura da árvore, por meio de setas

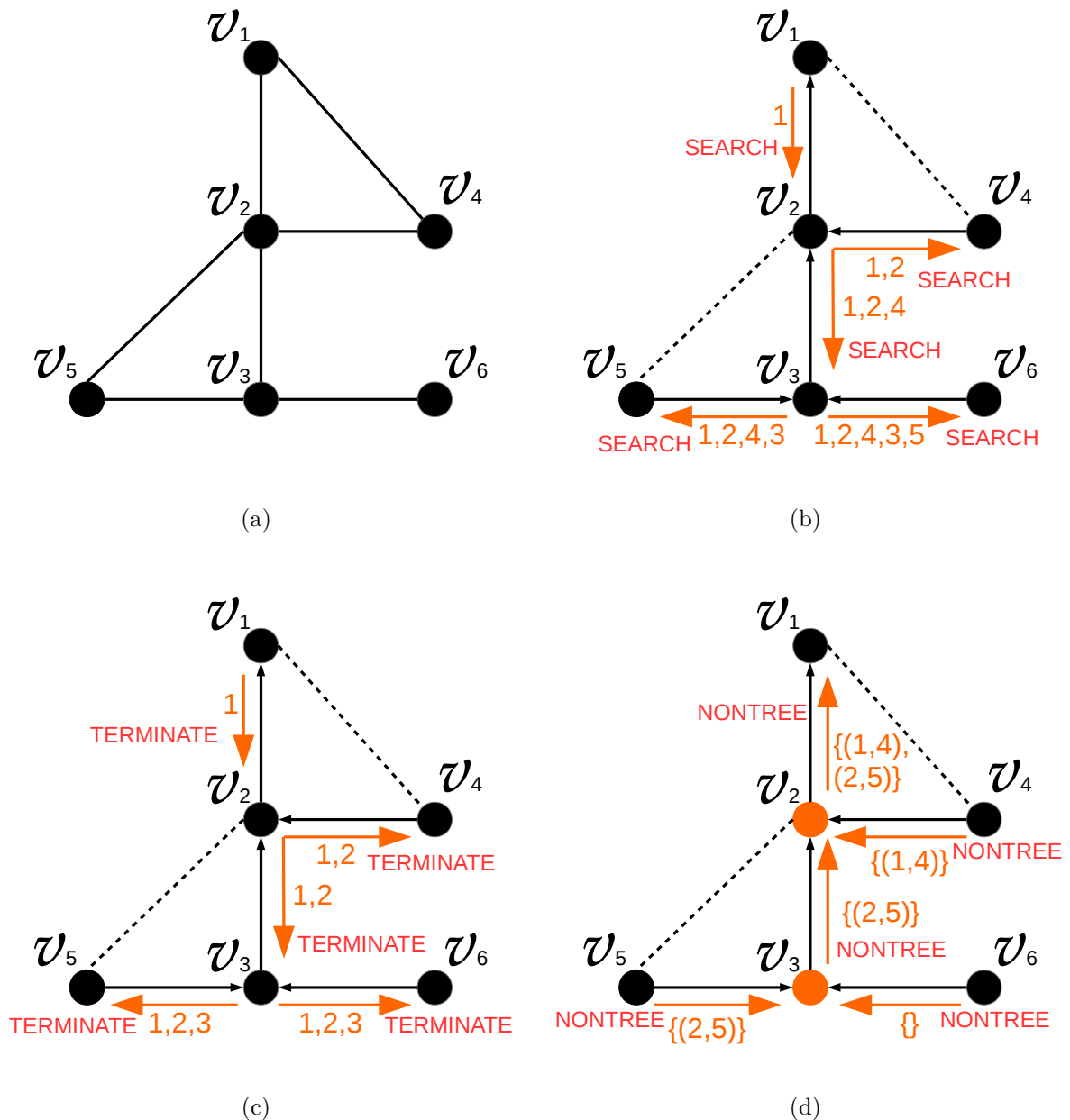


Figura 2.5: Exemplo para o algoritmo distribuído de Chaudhuri [14] para detectar articulações: (a) topologia da rede; (b) construção da árvore; (c) propagação dos ancestrais; (d) propagação dos ciclos e definição das articulações (nós  $v_2$  e  $v_3$ ).

nas arestas cuja direção representa o nó pai. Na Figura 2.5(c), os ancestrais de cada nó são definidos utilizando mensagens *TERMINATE* cuja propagação é realizada a partir da raiz. As mensagens são enviadas na ordem da busca em profundidade realizada anteriormente. Cada nó, ao receber esta mensagem, deve armazenar a informação, adicionar seu identificador e enviar aos seus descendentes. Na Figura 2.5(d), as arestas que não fazem parte da árvore (e que constituem ciclos) são propagados das folhas para as raízes utilizando mensagens *NONTREE*. Quando os nós recebem as mensagens, verificam se são ou não articulações. Os nós  $v_2$  e  $v_3$ , em destaque, são articulações. Note que  $v_3$  recebe de  $v_6$  uma mensagem informando que não existem ciclos naquele ramo e, portanto, identifica-se como articulação. O nó  $v_2$  recebe de  $v_3$  sua lista de arestas *nontree* e verifica que não existe, neste conjunto, uma aresta que liga a um de seus ancestrais. Assim, o nó  $v_2$  também determina que ele é uma articulação.

Turau [56] apresenta outra solução completamente distribuída para localizar articulações. Ela se difere das outras abordagens por utilizar a metodologia proposta em [54]. Esta alternativa permite que pontes e componentes 2-conectados sejam determinados. Assim, como no algoritmo original, a cada nó  $v_i$  é atribuído um valor *DFN* que determina a sequência de nós visitados em uma busca em profundidade e um valor  $L$  correspondente ao menor valor de *DFN* que pode ser obtido pelos nós por meio de um caminho para seus ancestrais. As articulações correspondem aos nós  $v_i$  que possuem filhos  $v_j$  tal que  $L(v_j) \geq DFN(v_i)$ . Estes valores são obtidos por mensagens especiais em um processo similar à proposta de Chaudhuri [14]. Os autores argumentam que a solução é ótima e utiliza  $O(|V|)$  unidades de tempo e mensagens para determinar as articulações, as pontes e os componentes 2-conectados. O algoritmo utiliza, no máximo,  $4 \cdot |V|$  mensagens.

A Tabela 2.1 apresenta um resumo dos algoritmos para a localização de articulações em redes sem fio. A coluna “Tipo” corresponde ao processamento utilizado: centralizado ou distribuído. Note que os algoritmos centralizados são caracterizados pela aplicação do conhecimento global da topologia, enquanto os algoritmos distribuídos utilizam conhecimento localizado a  $k$ -saltos ou requerem apenas o conhecimento dos nós vizinhos diretamente conectados. A coluna “Conhecimento” determina o tipo de informação/premissa empregada pelo algoritmo. A coluna “Tempo” representa o custo computacional, em relação a tempo. Nos algoritmos distribuídos, a complexidade corresponde ao tratamento das mensagens. A coluna “Mensagens” apresenta o custo computacional em relação a quantidade de mensagens necessárias. Neste caso, para os algoritmos centralizados, deve existir previamente uma coleta de informações sobre topologia que pode ser realizada com  $O(|V|^2)$  mensagens, onde cada nó informa qual o conjunto de arestas que possui [24].

Além das características resumidas na Tabela 2.1, deve-se notar que, apesar do baixo custo do algoritmo localizado de Jorgic *et al.* [24] (resultado da redução no tamanho da

Tabela 2.1: Mecanismos para localização de articulações em redes sem fio.

Mecanismo	Tipo	Conhecimento	Tempo	Mensagens
Goyal <i>et al.</i> [20]	Centralizado	Global	$O( V  +  E )$	$O( V ^2)$ *
Khelifa <i>et al.</i> [25]	Centralizado	Global	$O( V  +  E )$	$O( V ^2)$ *
Jorgic <i>et al.</i> [24]	Distribuído	$k$ -saltos	$O( V )$	$O(k \times (\Delta(G))^2)$ *
Chaudhuri [14]	Distribuído	1-salto	$O( V )$	$O( V )$
Turau [56]	Distribuído	1-salto	$O( V )$	$O( V )$

\* O trabalho original não informa o custo computacional.

entrada), esta é a única solução que produz falsos positivos (nós que não são articulações mas são detectados como tal). Os autores não determinam o custo, em termos de mensagens, para os algoritmos centralizados e o distribuído a  $k$ -saltos. É possível representar o grafo, em um nó qualquer da rede, quando todos ou parte dos nós enviam as informações sobre seus enlaces. Esta informação é encaminhada por nós intermediários até o destino, resultando em quantidade quadrática de mensagens [24]. Assim, o número de mensagens para os algoritmos centralizados pode ser estimado em  $O(|V|^2)$ . De forma similar, o número de mensagens para o algoritmo localizado a  $k$ -saltos é  $O(k \times (\Delta(G))^2)$ . Os algoritmos completamente distribuídos requerem  $4 \cdot |V|$  mensagens no total. Para os algoritmos centralizados, a identificação das articulações é armazenada na memória do nó que executa o algoritmo. Pode ser necessário propagar tal informação para que uma ação seja realizada durante a mitigação dos efeitos da presença de nós críticos. Nos algoritmos distribuídos, as informações são mantidas de forma distribuída e, portanto, cada nó possui em sua memória um conjunto de variáveis locais que inclui uma responsável por indicar se o nó é ou não uma articulação.

## 2.3 Mitigando os Efeitos da Presença de Articulações

A informação sobre a existência de articulações em redes sem fio foi utilizada de diferentes formas para garantir a conectividade ou para aumentar o tempo de vida da rede [2, 6, 7, 20, 25, 45]. Estas soluções incluem a implantação de novos nós (ou mudança da densidade), ativação de nós previamente dispostos em modo inerte, uso da mobilidade coordenada de nós e a redução do consumo de energia pela agregação de pacotes. Esta seção apresenta uma revisão da literatura sobre mecanismos que mitigam os efeitos causados pela presença de articulações em redes *ad hoc* sem fio.

Khelifa *et al.* [25] apresentam uma solução que, após detectar as articulações, reorganiza a rede para melhorar sua conectividade. A solução presume a existência de nós inertes, previamente implantados, que são chamados de “nós redundantes”. Além disso, devem existir nós móveis. A reorganização proposta consiste em: (i) localizar as articula-

ções; (ii) ativar um nó redundante próximo à articulação, se existir; e (iii) caso não exista nós redundantes próximo à articulação, o algoritmo seleciona um nó móvel e redundante de um vizinho e o movimenta para uma coordenada próxima à articulação. A Figura 2.6 apresenta um exemplo para esta solução. Na Figura 2.6(a) ilustra o grafo que representa o estado inicial da rede. Os nós redundantes estão representados em branco, enquanto as articulações  $v_4$  e  $v_7$  estão em cor cinza. Na Figura 2.6(b), o nó inerte próximo à articulação  $v_7$  é ativado, e um nó inerte, próximo a  $v_6$  é movido para as proximidades da articulação  $v_4$  e ativado. Note que os nós  $v_4$  e  $v_7$  deixam de ser articulações na nova topologia da rede. Os autores demonstram que, com um número suficiente de nós inertes e móveis, é possível eliminar a presença de articulações em uma rede *ad hoc* sem fio.

Goyal e Caffery [20] propuseram a postergação do particionamento da rede utilizando um conjunto de nós móveis, chamados auxiliares. Inicialmente, os enlaces críticos são identificados por meio de uma versão modificada do algoritmo DFS que mapeia a quantidade potencial de componentes. Em seguida, as articulações solicitam um nó auxiliar, que é movimentado para sua proximidade, reforçando a conectividade entre os componentes. A mensagem utilizada contém a informação de localização para onde o nó auxiliar deve se mover e a criticidade da articulação em termos do tamanho da partição. O trabalho não define como a posição dos nós pode ser obtida e foram considerados apenas posições em uma área quadrada no plano cartesiano. A propagação da mensagem é realizada por difusão e é limitada a 2 saltos para evitar o consumo excessivo de recursos. Dados de simulação apontam que, aproximadamente, 40% dos nós de articulação perdem esta

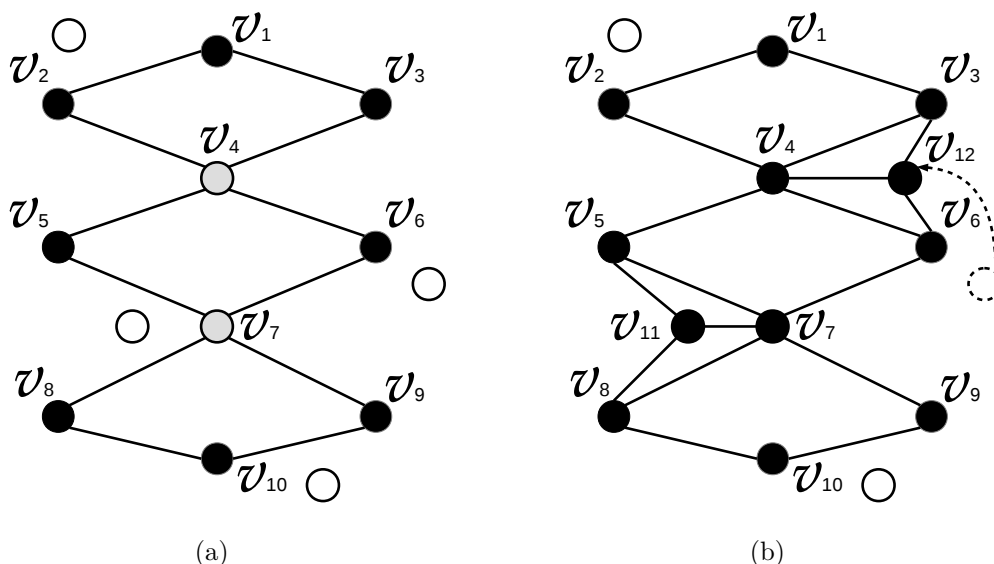


Figura 2.6: Reorganização da rede utilizando o mecanismo proposto por Khelifa *et al.* [25]: (a) grafo que representa a rede implantada; e (b) ativando nós inertes e movendo nós para regiões próximas a articulação.



característica ao utilizar esta solução.

Quando os dispositivos da rede utilizam energia proveniente de bateria, os nós estão suscetíveis à indisponibilidade em função do desvanecimento da energia remanescente, principalmente em função da quantidade de pacotes transmitidos e encaminhados. Em especial, nas redes de sensores, os dispositivos normalmente utilizam uma bateria não substituível [25, 59]. O consumo de energia desbalanceado acarreta na indisponibilidade de nós dependendo de características como sua localização ou a escolha do algoritmo de roteamento [59]. Além disso, as articulações (e conseqüentemente as pontes) são o único caminho entre nós de componentes distintos [53]. Assim, quando as articulações surgem, a rede tende a se tornar desconectada. Yong e Hu [59] destacam que isso pode ocorrer em condições onde os nós da rede tenham, em média, até 90% da energia inicial. Estas premissas constituem a base para uma solução, apresentada a seguir, que compõe as contribuições obtidas na pesquisa desenvolvida para esta tese.

Com base nas características descritas acima, Afonseca *et al.* [2] utilizaram a agregação de dados para reduzir o consumo de energia nas articulações em RSSF. A solução requer que cada nó possa verificar se é ou não uma articulação. Esta informação pode ser obtida por meio dos algoritmos de Chaudhuri [14] e Turau *et al.* [56]. O Capítulo 4 apresenta uma alternativa distribuída para localizar as articulações. Ao executar o DDFS para determinar as articulações, o nó sorvedouro deve ser a raiz da árvore. Assim, a partir de qualquer nó, é possível determinar a direção para o sorvedouro. Cada articulação informa aos seus vizinhos a  $k$ -saltos, em direção contrária ao sorvedouro, que eles devem realizar agregação. A agregação de dados é realizada concatenando os dados de pacotes disjuntos de tal forma que o volume dos cabeçalhos seja reduzido. Para controlar o atraso, é estipulado um limite de tempo que cada pacote pode aguardar no *buffer* até que outro chegue. Os pacotes são agregados até o limite máximo de transferência. Para avaliar a solução, foi utilizado um modelo de energia que considera a quantidade de dados recebidos e enviados. Este mecanismo de agregação permitiu a redução no consumo de energia das articulações e postergou o particionamento da rede. A Figura 2.7 apresenta como a agregação foi aplicada na proposta. A Figura 2.7(a) ilustra a topologia da rede. O nó identificado como sorvedouro inicia o processo de identificação das articulações. O nó em cor cinza é uma articulação e esta solicita aos vizinhos a 1-salto para realizarem a agregação de dados. A área destacada do grafo representa os nós que terão os dados agregados. Note que os nós de agregação podem incluir dados no *buffer* para a agregação. A Figura 2.7(b) exemplifica a agregação dos dados. Neste exemplo, dois pacotes são concatenados. O cabeçalho inicial contém as informações necessárias para encaminhar os dados para o sorvedouro. Cada segmento de dados contém um cabeçalho de agregação dentro do pacote. Lembre-se que todos os dados são direcionados para o nó sorvedouro,

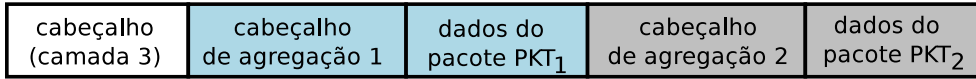
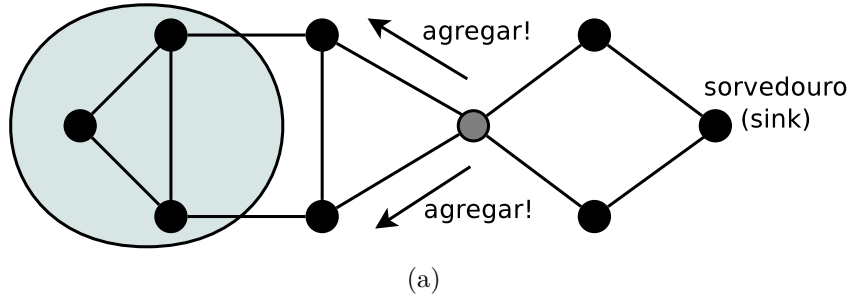


Figura 2.7: Exemplo da redução do consumo de energia nas articulações utilizando agregação e dados: (a) agregação de dados a 1 salto; (b) exemplo da agregação de dados para dois pacotes,  $PKT_1$  e  $PKT_2$ .

então a informação sobre o destino dos pacotes é suprimida no cabeçalho de agregação. De forma semelhante, todos os dados replicados podem ser suprimidos.

Para validar a técnica proposta em [2], o algoritmo foi submetido a um simulador de eventos discretos. Foram utilizadas topologias aleatórias com um único nó sorvedouro. Após a localização dos elementos críticos, os grafos com articulações foram submetidos a uma carga poissoniana com taxa  $\lambda = 10$  pacotes por segundo. Os pacotes têm, em média, 10 bytes de *payload* e mais 24 bytes de cabeçalho para as camadas de rede e transporte. As camadas física e de enlace utilizam um total 15 *bytes*, correspondente ao padrão IEEE 802.15.4 operando a 2.4Ghz, a uma taxa de 250Kbps e com endereçamento de 16 *bits*. O tempo máximo de retenção dos pacotes no buffer foi definido em  $\delta = 100ms$ . Foram analisados os cenários em que a agregação: (i) não é realizada, (ii) é realizada na articulação ( $k = 0$ ) e (iii) é realizada a um salto da articulação ( $k = 1$ ). A Tabela 2.2 apresenta o volume de dados trafegados na rede. O volume de informação útil é o mesmo para os três casos. Podemos notar uma redução de 14,91% quando a agregação é realizada na articulação e 28,48% para a agregação a um salto. A Tabela 2.3 sumariza o consumo

Tabela 2.2: Volume de dados trafegados na rede.

	Total	Cabeçalhos	Informação útil	Redução
Sem agregar	985.800	799.800	186.000	—
$k = 0$	838.800	652.800	186.000	14,91%
$k = 1$	704.967	518.967	186.000	28,48%

de energia na articulação. Ao agregar na articulação, a redução é de, aproximadamente, 20%, enquanto o consumo reduziu em, aproximadamente, 30% quando a agregação é realizada a um salto. Os dados de simulação apontam que a agregação de dados, quando

Tabela 2.3: Gasto médio de energia ( $mJ$ )

Consumo Energia	Sem Agregar	$k = 0$ (redução)	$k = 1$ (redução)
Total na RSSF	1.380.128	1.174.320 (14,91%)	986.953 (28,48%)
Total na Articulação	637.325	516.050 (19,02%)	440.971 (30,90%)
Total a Um Salto	644.735	596.235 ( 7,50%)	448.194 (30,48%)

aplicada de forma conveniente, é uma importante alternativa para aumentar o tempo de vida da rede.

A Tabela 2.4 resume os mecanismos para mitigar os efeitos da presença de articulações em redes *ad hoc* sem fio. A coluna “Mobilidade” indica se o mecanismo proposto requer a existência de características especiais nos sensores para sua mobilidade. A coluna “Nós extras”, indica se o mecanismo requer a existência de nós adicionais antes ou depois do particionamento. A coluna “Agregação” indica se a agregação de dados é utilizada. A coluna “Tipo” indica se o mecanismo proposto é centralizado ou distribuído.

Tabela 2.4: Mecanismos para mitigar os efeitos da presença de articulações.

Solução	Mobilidade	Nós extras	Agregação	Tipo
Khelifa <i>et al.</i> [25]	Sim	Sim	Não	Centralizado
Goyal <i>et al.</i> [20]	Sim	Não	Não	Distribuído
Afonseca <i>et al.</i> [2]	Não	Não	Sim	Distribuído

Podemos observar que o uso de agregação é a solução que apresenta o menor investimento, já que não necessita de informações sobre a localização geográfica dos nós, não requer nós sobressalentes e não necessita de nós móveis. Porém, ao realizar a agregação, os pacotes de dados podem sofrer atrasos. Para as soluções que requerem mobilidade, devemos observar que esta característica deve estar disponível em um subconjunto dos nós, mesmo que de forma limitada. Além disso, existem problemas inerentes a coordenação no deslocamento dos nós, em relação ao alto custo no consumo de energia para oferecer a mobilidade e a necessidade de prover mecanismos para obter informações sobre localização (como o sistema de posicionamento global). O uso de nós inertes é uma opção que, se devidamente planejada por meio de parâmetros como a densidade, pode oferecer resiliência à rede. Neste caso, são necessários mecanismos para ativar estes nós utilizando a comunicação sem fio. Técnicas como ciclos de ativação e inativação de nós são alternativas.

## 2.4 Considerações

Este capítulo apresentou uma base teórica sobre articulações e pontes e um conjunto de técnicas para localizar e mitigar os efeitos de sua presença em redes sem fio. A modelagem do surgimento destes elementos torna possível definir parâmetros de implantação. O uso de técnicas que ajustam a potência de transmissão dos nós, como o CT, reduzem a quantidade de arestas da rede e podem implicar na presença de elementos críticos. Porém, a conectividade da rede pode ser incrementada gradualmente para obter uma topologia semelhante com a menor quantidade possível de elementos críticos. A localização dos elementos críticos em redes *ad hoc* sem fio tem como base o algoritmo DFS sobre o conhecimento local ou global da topologia da rede. Nos algoritmos que utilizam informação global, o custo de comunicação para obter conhecimento da topologia e computar os elementos críticos são um desafio. As alternativas localizadas são distribuídas por natureza e, apesar de utilizarem algoritmos semelhantes, são suscetíveis a falsos positivos em função do conhecimento restrito. A aplicação do DDFS é uma alternativa que não requer a coleta de informações sobre a topologia e emprega um conjunto de mensagens para criar uma árvore em memória e determinar os elementos críticos. A mitigação dos efeitos da presença de elementos críticos tem como base a agregação de dados, a mobilidade de nós ou o uso de nós inertes. A agregação de dados orientada à presença de articulações aumenta o tempo de vida da articulação e, conseqüentemente, da rede. O uso de nós móveis e inertes também aumenta o tempo de vida da rede, porém nem sempre é possível ou viável. Considerando o estudo realizado, pode-se elencar como possibilidades de pesquisa e desenvolvimento:

- modelos analíticos e empíricos para prever surgimento de elementos críticos em redes *ad hoc* sem fio e verificar a relação da presença de articulações em regiões específicas, em especial, considerando o fenômeno chamado efeito de borda, em que o grau dos nós nesta região é não-linear [31, 43];
- técnicas para garantir a biconectividade por meio de enlaces cooperativos (criados com o Comunicação Cooperativa) que vão além do raio máximo de transmissão individual;
- técnicas para classificar articulações, por exemplo, em função do tamanho dos componentes ligados por ela e priorizar a mitigação dos efeitos causados por sua presença;
- técnicas para recuperação de conectividade que se empregam elementos críticos para prever a segmentação, sobretudo utilizando a Comunicação Cooperativa;

Neste trabalho, a informação sobre a existência de elementos críticos será utilizada para prever a segmentação da rede. A recuperação da conectividade é obtida pela ampliação do alcance dos nós por meio de um sistema MIMO virtual, chamado Comunicação Cooperativa. O Capítulo 3, que segue, apresenta uma revisão bibliográfica sobre esta técnica e apresenta trabalhos relacionados ao incremento de conectividade em redes *ad hoc* sem fio.

# Capítulo 3

## Comunicação Cooperativa

Em comunicação de dados, “diversidade” é uma técnica que consiste em utilizar múltiplos caminhos para os sinais, cada um com desvanecimento independente, tornando possível a melhoria na performance e na confiabilidade [38, 55]. Tse e Viswanath [55] afirmam que a diversidade pode ser obtida sobre o tempo, a frequência e o espaço. Na diversidade sobre o tempo, os símbolos são transmitidos em diferentes instantes aumentando a probabilidade de decodificar cada símbolo, já que o desvanecimento dos múltiplos sinais é diferente. Na diversidade sobre frequência, os nós podem utilizar a capacidade de seleção de frequência para enviar as múltiplas cópias dos símbolos. A diversidade espacial consiste em disponibilizar mais de uma antena em cada nó. Dispostas com uma distância suficiente, as antenas permitem a transmissão de múltiplas cópias de um mesmo símbolo. A Comunicação Cooperativa (CC) é uma técnica de diversidade em que nós portadores de uma única antena possam obter os benefícios da diversidade espacial ao transmitirem, simultaneamente, um mesmo sinal. Esta técnica, ao mitigar o desvanecimento de sinal, também permite aumentar o alcance dos nós.

Este capítulo apresenta uma revisão sobre o uso da diversidade espacial de forma distribuída e sua aplicação nas redes *ad hoc* sem fio. A Seção 3.1 expõe os conceitos básicos da diversidade espacial e da CC. A Seção 3.2 contém um modelo de comunicação que permite que a origem envie dados, cooperativamente, para um nó além de  $R_{MAX}$  utilizando um ou mais auxiliares. Apesar de simples, o modelo facilita a implementação, simulação, análise e compreensão de resultados em processos de simulação. Modelos semelhantes foram utilizados em trabalhos relacionados permitindo a comparação com a solução proposta [61, 62]. Em seguida, a Seção 3.3 exibe uma revisão de trabalhos que utilizam a CC como mecanismo de CT para aumentar a conectividade da rede. Por fim, a Seção 3.4 resume o capítulo e discute o uso da Comunicação Cooperativa na recuperação de conectividade.

### 3.1 Esquemas de Comunicação Cooperativa

Na diversidade espacial, múltiplas antenas são disponibilizadas na origem ou no destino. Dispostas em uma distância suficiente, as antenas proporcionam múltiplos caminhos que são estatisticamente diferentes [21]. A confiabilidade do canal aumenta com a existência de múltiplas cópias [55]. A diversidade espacial pode ser obtida tanto no transmissor quanto no receptor, por meio de diferentes arranjos [21]. O transmissor pode enviar os dados por meio de uma única antena, enquanto o receptor utiliza várias antenas para receber o sinal. De forma semelhante, o transmissor pode ser portador de múltiplas antenas, à medida que o receptor pode decodificar o sinal com uma única antena. O *Multiple-Input Multiple-Output* (MIMO), ou Múltiplas Entradas e Múltiplas Saídas, é uma técnica em que múltiplas antenas são embarcadas em transceptores modernos para obter a diversidade espacial, utilizadas tanto no transmissor quanto no receptor, para enviar ou receber dados em múltiplos caminhos. A Figura 3.1 representa um sistema MIMO, nela, são demonstrados dois nós com múltiplas antenas. O nó identificado com *TX* envia múltiplos sinais ao nó *RX*. O nó *RX* utiliza suas antenas para decodificar as múltiplas cópias do sinal. Em [21], os autores apresentam uma revisão teórica sobre MIMO e detalham modelos para o sinal recebido em cada antena e a probabilidade de *outage* (probabilidade de que o sinal recebido esteja abaixo do limiar requerido pelo destinatário) quando se aplica o modelo *Rayleigh* para o desvanecimento de sinal.

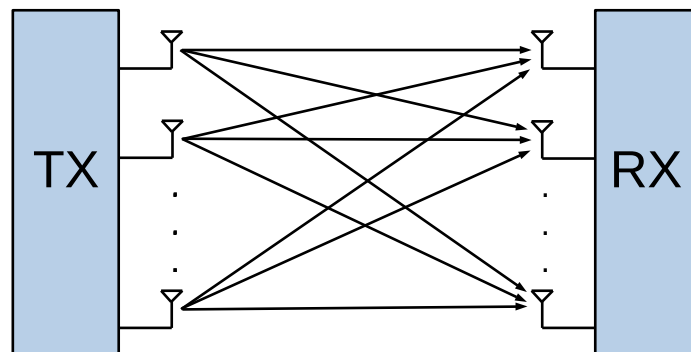


Figura 3.1: Comunicação usando MIMO.

Disponibilizar múltiplas antenas em sensores ou pequenos dispositivos é inviável [38]. Neste caso, é possível montar um sistema de antenas distribuído em que os nós cooperam para utilizar suas antenas independentes em um arranjo semelhante ao MIMO. Esta técnica é chamada de Comunicação Cooperativa (CC). Na CC, um conjunto de nós chamados auxiliares transmitem, cooperativamente, o mesmo sinal em conjunto com o nó de origem. A Figura 3.2 ilustra a CC: a origem e dois auxiliares enviam a mesma informação

para o nó de destino. Na CC, a origem deve enviar os dados para os auxiliares para que a informação seja replicada. O destino, por sua vez utiliza as características da camada física para combinar sinais parciais e obter a informação completa [49]. O resultado é um sistema MIMO virtual que combate efetivamente a atenuação de sinal ao transmitir cópias independentes de um mesmo dado por meio de vários nós em uma rede sem fio [38].

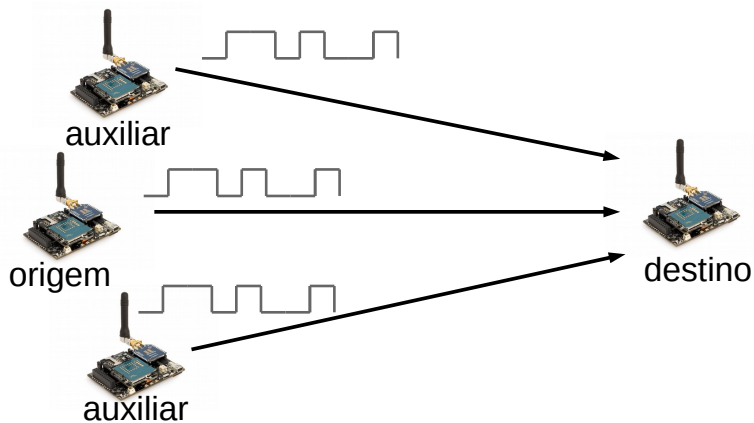


Figura 3.2: Exemplo de Comunicação Cooperativa.

Segundo [11, 29, 38], são dois os métodos básicos de cooperação: Amplificar e Encaminhar ou *Amplify and Forward* (AF) e Decodificar e Encaminhar ou *Decode and Forward* (DF). Estes métodos se diferem quanto à forma em que os nós auxiliares manipulam os dados ao encaminhá-los. No AF, cada nó auxiliar recebe uma versão do sinal que pode conter ruídos, o sinal é amplificado e retransmitido sem nenhum tipo de processamento, o destino combina as informações. A principal vantagem desta técnica é sua baixa complexidade, sobretudo na implementação. Porém, os erros recebidos pelo auxiliar são propagados para o destino. Laneman e Wornell [30] modelaram os eventos de *outage* para a cooperação de dois nós utilizando variáveis Gaussianas complexas e definem que um sinal não é corretamente recebido quando

$$\gamma_{s,d} + f(\gamma_{s,r}, \gamma_{r,d}) < t, \quad (3.1)$$

onde  $f(x, y) = [x^{-1} + y^{-1} + (xy)^{-1}]^{-1}$ ,  $\gamma_{i,j}$  indica o SNR de cada sinal enviado por  $i$  e recebido por  $j$ ,  $s$  indica a origem,  $d$  o destino,  $r$  o nó auxiliar e  $t$  é o limiar para o valor SNR, proporcional à taxa de transmissão, para que o destino decodifique o sinal. No DF, cada auxiliar decodifica o sinal recebido e retransmite a informação (semelhante ao encaminhamento de pacotes). Apesar de apresentar maior complexidade na implementação, este método evita a propagação de erros e se torna interessante quando o SNR é alto entre a origem e o auxiliar [30]. Um evento de *outage* na cooperação por DF é definido como



$$\gamma_{s,r} < t \quad \vee \quad \gamma_{s,d} + \gamma_{r,d} < t. \quad (3.2)$$

Note que, conforme as Equações 3.1 e 3.2, o evento de *outage* ocorre quando o sinal recebido pelo auxiliar não atinge o limiar ou, quando a soma dos sinais recebidos pelo destino não atingem o limiar.

A Figura 3.3 exemplifica os dois principais métodos de cooperação, AF e DF. Nesta figura, um nó de origem envia dados a um nó de destino utilizando um único auxiliar. Na Figura 3.3(a), o nó auxiliar recebe o sinal, amplifica e o retransmite. Observe que o nó auxiliar pode obter o sinal durante a comunicação entre a origem e o destino. Como o sinal é propagado sem nenhum tipo de processamento, os dados são enviados com os erros provenientes do desvanecimento do sinal e interferência. Na Figura 3.3(b), o nó auxiliar recebe o sinal, decodifica e retransmite. Algoritmos de detecção e correção de erros podem ser utilizados. Um atraso é adicionado ao tempo de transmissão em função da decodificação, porém os erros recebidos pelo auxiliar não são propagados ao destino.

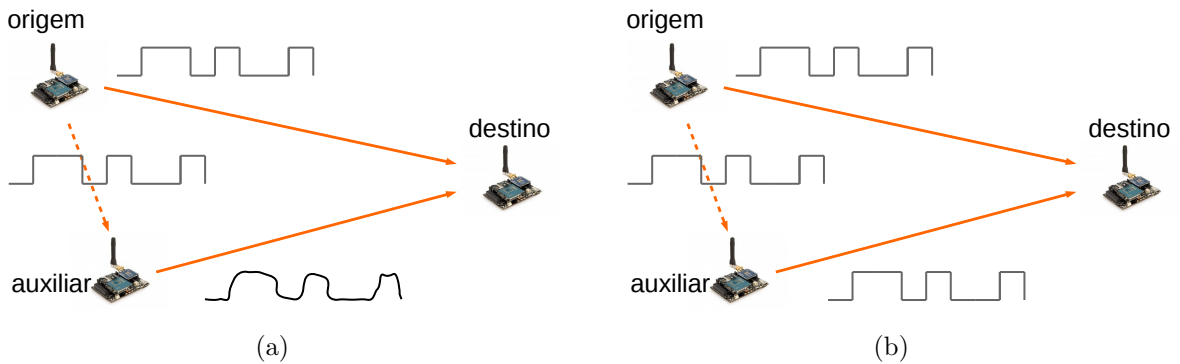


Figura 3.3: Métodos de cooperação: (a) Amplificar e Encaminhar (ou AF) e (b) Decodificar e Encaminhar (ou DF).

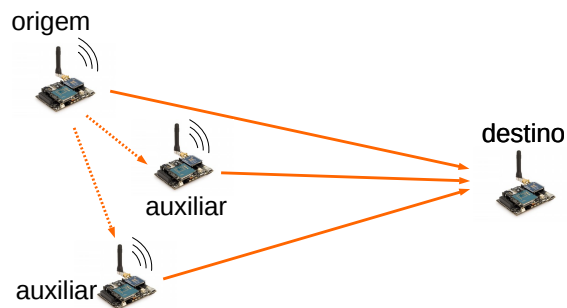


Figura 3.4: Comunicação Cooperativa em única etapa.

Além dos métodos de cooperação, a Comunicação Cooperativa poder ser realizada em uma ou duas fases [21]. Na comunicação de fase única, o nó auxiliar escuta, ocasional-

mente, o sinal e, imediatamente envia os dados para o destino utilizando um dos métodos de cooperação listados acima. Pode-se inferir que o nó auxiliar utiliza da característica de difusão do meio para receber o sinal proveniente da origem. A Figura 3.4 ilustra a cooperação em única fase. Neste exemplo, dois nós auxiliares escutam o sinal e o retransmitem. A comunicação em duas fases é dividida em fase de coordenação e fase de cooperação. Na fase de coordenação, a origem envia os dados para os nós auxiliares. Para que os dados sejam enviados aos nós auxiliares, a origem utiliza da natureza do meio sem fio e realiza uma difusão [62]. Em seguida, na fase de cooperação, a origem e seus auxiliares enviam os dados, simultaneamente, para o destino. A Figura 3.5 exemplifica a CC em duas fases. A Figura 3.5(a) corresponde à fase de coordenação em que a origem envia os dados para um único auxiliar. Em geral, a origem utiliza uma potência de transmissão necessária apenas para alcançar o auxiliar mais distante. A Figura 3.5(b) corresponde à fase de cooperação em que a origem e seu auxiliar enviam os dados para o destino. Os nós devem utilizar uma potência de transmissão que, combinadas, permitem que o destino consiga decodificar o sinal. Ao utilizar duas fases, a origem transmite a mesma informação duas vezes aumentando assim o seu consumo de energia em relação à cooperação em fase única. Além disso, como a origem participa na transmissão da segunda fase, a energia individual dos nós pode ser diluída entre ela e os auxiliares.

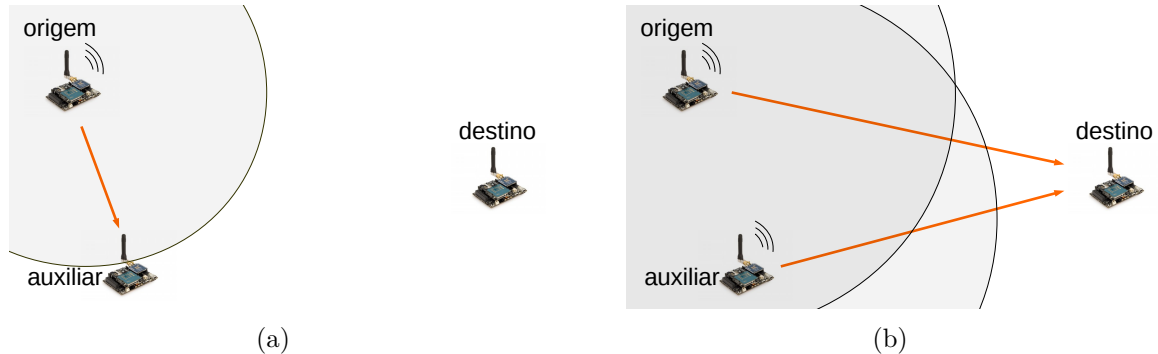


Figura 3.5: Comunicação Cooperativa em duas etapas: (a) fase de coordenação; e (b) fase de cooperação.

Um dos problemas inerentes à Comunicação Cooperativa é a seleção dos nós auxiliares, que pode ser realizada de forma centralizada ou distribuída [10]. Na seleção centralizada, um nó é responsável por coletar as informações da topologia da rede e posição dos nós, decidir quais nós serão utilizados como auxiliares e enviar esta informação às origens. Em geral, é necessário um nó com maior capacidade como uma estação base para fazer a seleção dos auxiliares. Na seleção distribuída, cada nó coleta um conjunto de informações da topologia e escolhe, individualmente, seus auxiliares [1]. Em geral os nós auxiliares são selecionados com o objetivo de aumentar a vazão individual do nó ou a redução no custo

de energia. Em [15], os autores apresentam uma técnica para seleção de auxiliares que considera a energia residual dos nós. Em Jamal e Mendes [23], os autores apresentam uma taxonomia e uma revisão das principais técnicas da literatura para a seleção de auxiliares.

Outra questão importante, durante a cooperação, é a potência de transmissão utilizada pela origem e pelos auxiliares. A potência deve ser suficiente para que o destino consiga decodificar o sinal combinado após o desvanecimento e interferências. Além disso, é necessário atingir um determinado desempenho, como, por exemplo, trabalhar fora da região de *outage*, que depende diretamente da capacidade do canal em *bits/s*. Em [21] são apresentados modelos para a definição da potência de transmissão na comunicação cooperativa. Zhu *et al.* [62] simplificam o problema distribuindo igualmente a potência entre os nós.

A CC, além de combater o desvanecimento do sinal, também pode ser utilizada para permitir que os nós transmitam informação para destinos além do raio máximo de transmissão individual [62]. As Equações 3.1 e 3.2 demonstram que o sinal pode ser decodificado, quando a soma das potências recebidas é superior ao limiar da potência de decodificação. Assim, espera-se que um nó de destino fora do raio de comunicação pode, efetivamente, decodificar quando a potência utilizada pela origem e seus auxiliares, ao transmitir os dados, seja suficiente para a decodificação. A Figura 3.6 exemplifica o cenário em que a Comunicação Cooperativa é utilizada pela origem  $v_i$  para enviar dados a um nó de destino  $v_j$  que está fora do raio máximo de transmissão. Nesta figura, o raio de comunicação máximo da origem é representado pela área em cinza claro e, como pode ser observado, ele possui três vizinhos. Os vizinhos diretamente conectados, com a potência máxima de transmissão, são candidatos a auxiliares na Comunicação Cooperativa. Considere que a comunicação é realizada em duas etapas e que algum método arbitrário selecionou apenas um nó como auxiliar. Neste caso, a área em cinza escuro representa o raio atingido pela origem ao utilizar a potência de transmissão necessária para alcançar o auxiliar. Esta área corresponde à potência de transmissão utilizada na primeira fase. Em seguida, o nó auxiliar e a origem enviam os dados, simultaneamente, como uma potência que possibilite a decodificação do sinal ao destino.

Diversos autores justificam que existe um equilíbrio, ao utilizar CC, em relação ao uso de energia e à eficiência da rede [11, 30, 38]. Nosratinia *et al.* [38] destacam que o consumo de energia da rede é maior na Comunicação Cooperativa já que os nós devem utilizar o transceptor para enviar seus dados e de outros nós e, mesmo assim, espera-se que o consumo de energia dos nós seja reduzido. Este comportamento é explicado pela redução na potência individual dos nós ao cooperarem na transmissão dos dados. Os autores argumentam que é possível aumentar a eficiência no uso do espectro já que a taxa de transmissão individual dos nós também aumenta.

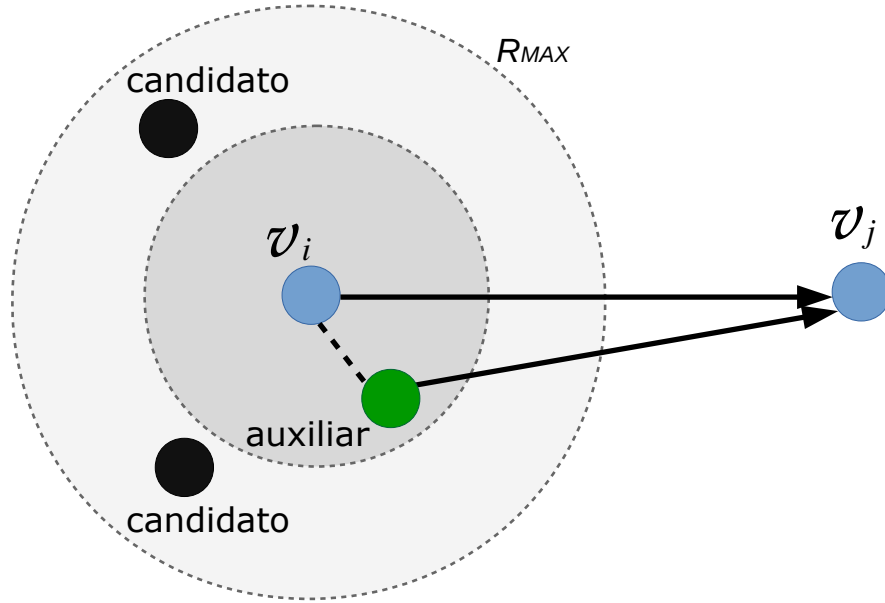


Figura 3.6: Uso da Comunicação Cooperativa para extrapolar o raio de comunicação máximo.

### 3.2 Modelo de Comunicação Cooperativa

Apesar da possibilidade de utilizar o *Modelo de Propagação no Espaço Livre* ou o *Modelo de Dois Raios*, é comum a utilização de um modelo genérico para fazer avaliações analíticas e simulação [21]. O modelo de Comunicação Cooperativa apresentado a seguir é baseado nos trabalhos de Yu et al. [60] e Zhu et al. [62]. Nele, todos os nós podem transmitir dados ou atuarem como agentes cooperativos, retransmitindo dados. A comunicação entre pares de nós ocorre a múltiplos saltos incluindo arestas diretas ou cooperativas. Ao realizar a Comunicação Cooperativa, os nós que transmitem os dados serão referenciados como origem, enquanto seus vizinhos diretamente conectados são chamados de candidatos a auxiliares. Os nós selecionados para cooperação são chamados de nós auxiliares. A comunicação ocorre em duas etapas e a origem utiliza da natureza do meio sem fio para enviar os dados para todos os nós auxiliares em uma única transmissão por meio de difusão [62]. Os auxiliares são previamente definidos ou escolhidos a cada transmissão. A potência de transmissão necessária pela origem, em conjunto com os nós auxiliares, pode ser determinada de forma semelhante à comunicação direta, conforme a Equação 2.2. Desta forma, a comunicação completa entre os nós  $v_i$  e  $v_j$  pode ser obtida, com comunicação CC, se  $v_i$  transmite seu sinal juntamente com um conjunto de nós auxiliares  $H_{i,j}$  e o somatório das potências recebidas por  $v_j$ , ou  $P^r$ , satisfaz a Equação 3.3.

$$P^r = \sum_{v_k \in v_i \cup H_{i,j}} P_k (d_{k,j})^{-\alpha} \geq \tau \quad (0 \leq P_k \leq P_{MAX}), \quad (3.3)$$

onde:  $\alpha$  é o expoente de desvanecimento do sinal;  $d_{i,j}$  é a distância Euclidiana entre  $v_i$  e  $v_j$ ; e  $\tau$  é o valor mínimo para o SNR recebido por  $v_j$  para que este nó decodifique o sinal. Neste modelo, todos os nós utilizam a mesma potência de transmissão durante a segunda fase. Assim, a potência de transmissão individual pode ser definida por

$$P^t = \frac{\tau}{\sum_{v_k \in v_i \cup H_{i,j}} (d_{k,j})^{-\alpha}}. \quad (3.4)$$

Quando for possível definir um conjunto de auxiliares para a origem e a potência de transmissão individual para atingir um nó de destino qualquer, novas arestas são introduzidas no grafo e estas não podem utilizar os conceitos clássicos. Assim, são apresentadas algumas definições importantes semelhantes às aquelas propostas em [62]:

**Definição 3.2.1.** (*Enlace direto*): Um enlace direto  $\overline{v_i v_j}$  é uma aresta em  $E$  representando que o nó  $v_i$  pode transmitir dados para o nó  $v_j$  diretamente, isto é,  $P_i$  é tal que o nó  $v_i$  pode alcançar  $v_j$  quando  $P_i \leq P_{MAX}$ .

**Definição 3.2.2.** (*Conjunto de nós auxiliares*):  $H_{i,j}$  representa o conjunto de nós auxiliares de  $v_i$  em uma comunicação cooperativa com  $v_j$ . Todos os nós auxiliares necessários são vizinhos diretos de  $v_i$ , isto é,  $H_{i,j} \subseteq N(v_i)$ , onde  $N(v_i)$  é o conjunto de todos vizinhos diretos de  $v_i$ .

**Definição 3.2.3.** (*Enlace cooperativo*): Um enlace cooperativo, ou enlace-cc,  $\widetilde{v_i v_j}$  é uma aresta de  $E$  que representa que um nó  $v_i$  pode transmitir dados para  $v_j$  cooperativamente utilizando um conjunto de nós auxiliares  $H_{i,j}$ .

**Definição 3.2.4.** (*Enlace auxiliar*): Um enlace auxiliar é uma aresta entre  $v_i$  e um de seus nós auxiliares em  $H_{i,j}$ .

**Definição 3.2.5.** (*Topologia de rede*): A união de todos os enlaces diretos e dos enlaces cooperativos,  $\overline{E}$  e  $\widetilde{E}$ , respectivamente. De forma similar, o grafo de comunicação direta e o grafo de comunicação CC são denotados por  $\overline{G} = (V, \overline{E})$  e  $\widetilde{G} = (V, \widetilde{E})$ , respectivamente. Assim, o grafo  $G(V, E)$  é composto pelos vértices  $V$  e pelas arestas  $E = \overline{E} \cup \widetilde{E}$ . Seguindo a notação, note que, se  $v_i v_j \in \overline{E}$ , então:  $v_i v_j = \overline{v_i v_j}$  se  $v_i v_j$  é um enlace direto e  $v_i v_j = \widetilde{v_i v_j}$  se  $v_i v_j$  é um enlace cooperativo.

**Definição 3.2.6.** (*Peso do enlace direto*): O peso de um enlace direto  $\overline{v_i v_j}$  é definido como:

$$w(\overline{v_i v_j}) = \tau d_{i,j}^\alpha. \quad (3.5)$$

**Definição 3.2.7.** (*Peso do enlace cooperativo*): O peso de um enlace CC  $\widetilde{v_i v_j}$  é definido como:

$$w(\widetilde{v_i v_j}) = w_d(H_{i,j}) + (|H_{i,j}| + 1)w_{CC}(H_{i,j}), \quad (3.6)$$

onde:

- $|H_{i,j}|$ : é o número de elementos em  $H_{i,j}$ ;
- $w_d(H_{i,j}) = \left( \frac{\tau}{\max_{v_k \in H_{i,j}} (d_{i,k})^{-\alpha}} \right)$ : é a potência de transmissão utilizada pelo nó  $v_i$  para comunicar com o nó mais distante em  $H_{i,j}$  - no exemplo da Figura 3.6, a origem utiliza a potência necessária para atingir apenas o vizinho mais próximo que foi previamente selecionado como auxiliar;
- $w_{CC}(H_{i,j}) = \left( \frac{\tau}{\sum_{v_k \in v_i \cup H_{i,j}} (d_{k,j})^{-\alpha}} \right)$ : é a potência de transmissão individual do  $v_i$  e de seus auxiliares em  $H_{i,j}$  para comunicar cooperativamente com  $v_j$  - na definição, esta potência é multiplicada por  $(|H_{i,j}| + 1)$ , que corresponde ao número total de nós transmitindo na segunda fase.

De acordo com as Equações 2.2 e 3.3, a relação  $\max(w_d(H_{i,j}), w_{CC}(H_{i,j})) \leq P_{MAX}$  deve ser satisfeita para existir um enlace cooperativo. O peso de um enlace cooperativo consiste na soma do custo de comunicação das etapas de coordenação e cooperação. O custo para a etapa de coordenação é equivalente a  $w_d(H_{i,j})$ , enquanto o custo individual dos nós para transmitir dados durante a fase de cooperação é  $w_{CC}(H_{i,j})$ .

O modelo apresentado, nesta seção, considera apenas se o destino consegue decodificar os sinais combinados. Não são modelados os atrasos, erros ou interferências. Assim, a comunicação pode ocorrer tanto no método de cooperação AF quanto no DF. Apesar da simplicidade, este modelo permite analisar as propriedades dos grafos, como, por exemplo, pesos de arestas diretas, cooperativas e caminhos entre nós. Modelos semelhantes foram utilizados em [36, 61, 62]. Neste trabalho, não foram analisados transmissões de pacotes ou consumo de energia (potência em função do tempo). O objetivo é, portanto, permitir a comparação das soluções em relação ao custo das arestas ou dos caminhos criados entre pares de nós.

Como discutido na seção anterior, o alcance dos nós pode ser ampliado em função da Comunicação Cooperativa. Esta técnica pode ser utilizada para incrementar a conectividade da rede, criando enlaces entre nós, cuja distância é maior que o raio de transmissão máximo na comunicação direta. Desta forma, a CC é utilizada como um mecanismo de Controle de Topologia que adiciona arestas cooperativas à rede. A Seção 3.3, a seguir, apresenta um conjunto de trabalhos que utilizam CC para melhorar a conectividade da rede.

### 3.3 Incremento de Conectividade Utilizando Comunicação Cooperativa

Tendo em vista a necessidade de manter a conectividade e prolongar a vida útil dos nós, várias técnicas voltadas a preservar ou mesmo reduzir o consumo de energia foram propostas [11, 33, 59, 61, 62]. Dentre estas, o Controle de Topologia (CT) tem sido amplamente utilizada, cujo objetivo consiste em controlar os enlaces de comunicação entre os nós, por meio do raio de transmissão, com o objetivo de manter alguma propriedade global, enquanto reduz o consumo de energia ou interferência [50]. Esta técnica permite a manutenção de diversas características da rede, ao passo que reduz a interferência e o consumo de energia por meio do ajuste de potência de transmissão dos nós.

Em geral, o CT é utilizado para remover arestas de maior peso (maior consumo de energia) de tal forma que a energia total consumida pela rede seja minimizada durante a operação [44]. Neste contexto, a Comunicação Cooperativa (CC) passou a ser utilizada, ao contrário das técnicas clássicas, para incrementar o número de arestas. A CC foi aplicada para reduzir o consumo de energia permitindo a comunicação por meio de enlaces cooperativos quando estes apresentam menor custo que enlaces diretos [11]. Neste cenário, quando o caminho entre origem e destino é considerado, os enlaces cooperativos podem se tornar ainda mais atraentes. Em especial, podem ser utilizados para conectar os componentes de rede quando estes não podem ser ligados por meio de enlaces diretos [59, 61, 62]. Em geral, esta alternativa é aplicada durante a implantação da rede devido ao alto custo computacional para calcular enlaces cooperativos viáveis. O resto desta seção apresenta um conjunto de técnicas que aplicam CC no Controle de Topologia. Soluções que estão diretamente relacionadas a este trabalho são abordadas com mais detalhes.

Cardei *et al.* [11] definem o problema do Controle de Topologia em Comunicação Cooperativa com o objetivo de reduzir o consumo de energia e demonstram que tal problema é NP-completo. Como proposta, os autores apresentam os algoritmos *Distributed Topology Control Using CC* (DTCC) e *Incremental Topology Control Using CC* (ITCC) que, aplicados a uma topologia conectada, pode reduzir o consumo total de energia. A topologia inicial, em que a solução é aplicada, resulta do CT tradicional como o *Minimum Spanning Tree* (MST) ou Árvore Geradora Mínima. No DTCC, dado um grafo conectado, os nós iniciam com a potência  $P_i^{MAX}$  necessária para alcançar seu vizinho com maior distância. Os nós conhecem a distância e a potência dos vizinhos a 2-saltos. A partir destas informações, os nós definem um incremento em sua potência de transmissão considerando a redução na potência total da rede. O incremento corresponde ao nível de contribuição do nó com a comunicação dos seus vizinhos a 2-saltos para reduzir a potência dos vizinhos a 1-salto. No algoritmo ITCC, cada nó  $v_i$  calcula a potência  $P_i^{MIN}$  para alcançar o nó

mais próximo e a potência  $P_i^{MAX}$  necessária para alcançar o vizinho mais distante. Em seguida, o algoritmo encontra a potência mínima de transmissão para o nó  $v_i$  para que seus vizinhos  $N(v_i)$  sejam completamente cobertos utilizando CC. Dados de simulação são apresentados para uma combinação dos algoritmos *DTCC* e *ITCC* com os algoritmos *Distributed Minimum Spanning Tree* (DMST) ou Árvore Geradora Mínima Distribuída e *Localized Minimum Spanning Tree* (LMST) ou Árvore Geradora Mínima Localizada. Como resultado, os autores demonstram que é possível reduzir em até 21% a potência de transmissão dos nós, no melhor caso, com as técnicas propostas.

Yu *et al.* [61] utilizam a CC como um mecanismo de CT cujo objetivo é ligar componentes disjuntos. Uma versão aprimorada do trabalho foi apresentada, posteriormente, em [60]. A solução proposta, chamada *Cooperative Bridges* ou *CoopBridges*, aumenta a conectividade da rede, enquanto reduz a potência de transmissão. O algoritmo possui cinco etapas: (i) construção dos componentes em  $G$  utilizando a potência máxima de transmissão; (ii) construção de um grafo  $G'$  que contém os vértices de  $G$  e uma seleção de candidatos para os enlaces CC bidirecionais entre os componentes; (iii) um grafo  $G''$  é obtido, ao considerar que cada componente de  $G'$  como um nó em  $V(G'')$  e cada aresta cooperativa entre os componentes de  $G'$  em  $E(G'')$ ; o número de arestas de  $G''$  é reduzido utilizando um algoritmo de árvore geradora mínima; as arestas de  $G'$  que não estão presentes em  $E(G'')$  são removidas; (iv) as arestas não eficientes, dentro dos componentes de  $G$ , são reduzidas utilizando, novamente, um algoritmo de árvore geradora mínima, à medida que as arestas que são utilizadas na primeira fase da CC (enlaces auxiliares) são mantidas; (v) os grafo  $G$  e  $G'$  são combinados resultando em um grafo que corresponde, portanto, às arestas de custo mínimo tanto para a comunicação direta quanto para a CC. A Figura 3.7 apresenta um exemplo para a solução de Yu *et al.* [61]. A Figura 3.7(a) mostra o grafo inicial. Na Figura 3.7(b) as áreas demarcadas representam os componentes de rede ligados por meio das linhas que correspondem aos enlaces cooperativos de menor custo. A Figura 3.7(c) representa o grafo resultante após a eliminação dos enlaces diretos de maior custo. A Figura 3.7(d) ilustra o grafo resultante após a combinação das arestas diretas e cooperativas de menor custo. Por meio de simulações, os autores demonstram que a solução proposta consegue aumentar a conectividade em 50% em relação ao DTCC [12]. Os autores argumentam que este resultado é obtido com um aumento tolerável na potência de transmissão. Para um fator de desvanecimento de sinal de  $\alpha = 2$ , a solução combinada com o DTCC requer um aumento na média da potência individual de 13% e quando aplicado sozinho, necessita de um aumento de 18%.

Ainda, na proposta de Yu *et al.* [61], foi demonstrado que o problema de seleção de auxiliares para a redução do consumo de energia é NP-Completo. Os autores apresentam dois mecanismos: um método exaustivo e uma heurística gulosa. A heurística, chamada



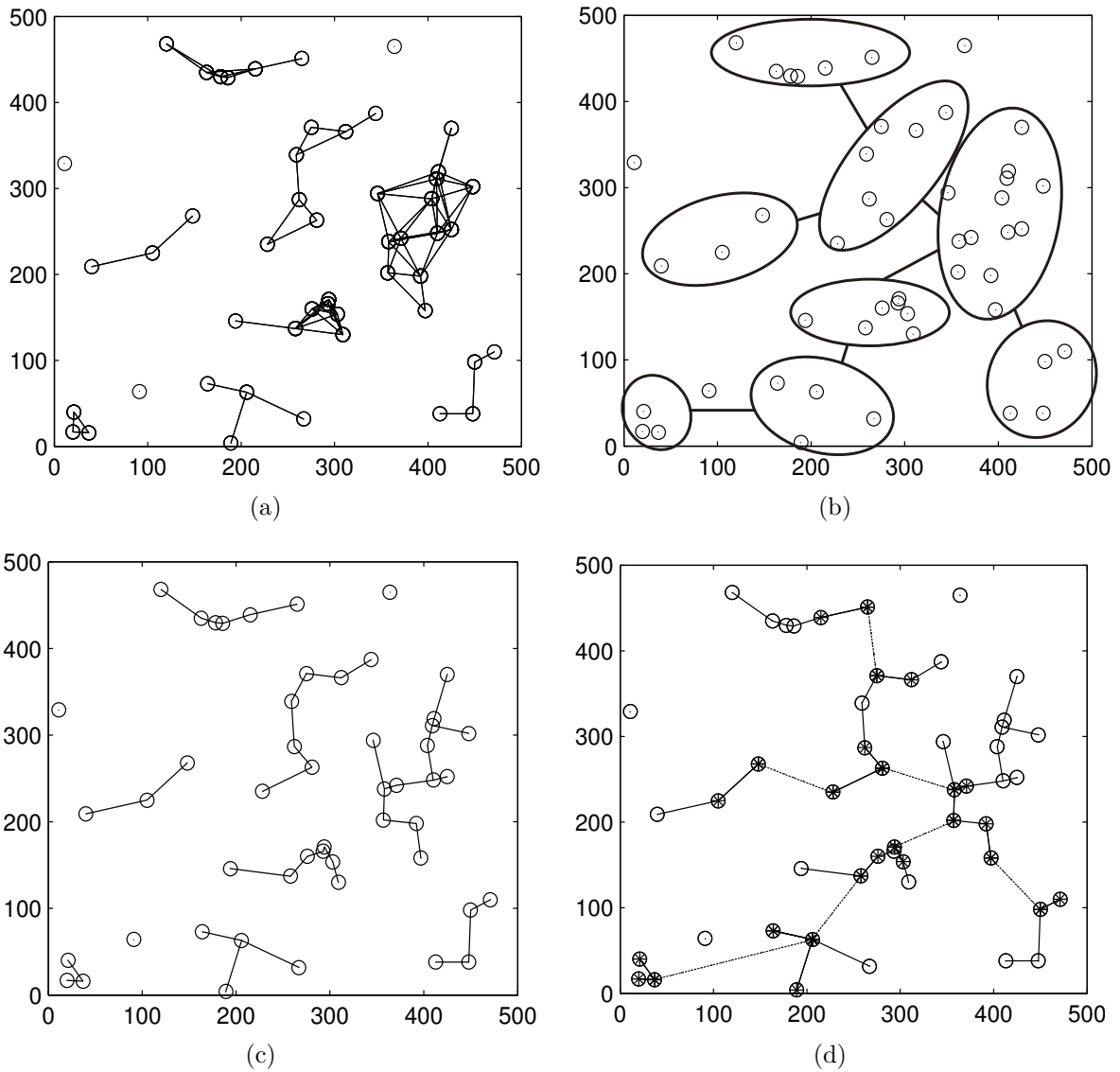


Figura 3.7: Exemplo do *CoopBridges* (Fonte: [61]): (a) grafo inicial; (b) enlaces resultantes após aplicar a árvore geradora mínima sobre os enlaces cooperativos que ligam os clusters; (c) enlaces resultantes após a aplicação da árvore geradora mínima nos enlaces diretos; (d) grafo resultante após a combinação dos enlaces diretos e dos enlaces cooperativos.

*Greedy Helper Set Selection* (GHSS) computa um conjunto de nós auxiliares em tempo polinomial. O algoritmo tem como entrada um nó de origem  $v_u$ ; um nó de destino  $v_v$ ; o conjunto de nós vizinhos da origem, denotado  $N(v_u)$ . As distâncias  $d_{u,v}$ ,  $d_{u,i}$ , e  $d_{v,i}$ , para  $v_i \in N(v_u)$ , devem ser conhecidas. A saída do algoritmo é uma estimativa de custo do enlace cooperativo  $\widetilde{v_u v_v}$ . Inicialmente o algoritmo ordena os nós em função de uma métrica que considera quanta energia a origem pode economizar ao utilizar aquele vizinho como auxiliar e a potência necessária para enviar os dados a ele. Em seguida, os nós são

adicionados até que seja possível estabelecer o enlace cooperativo. Por fim, os outros nós são acrescentados ao conjunto de auxiliares se não aumentarem a potência de comunicação. Os nós utilizam a mesma potência de transmissão durante a fase de cooperação. Desta forma, ao adicionar mais nós que o necessário, ao conjunto de auxiliares, a potência de transmissão individual é reduzida.

Zhu *et al.* [62] definem o problema *Energy-Efficient Topology Control Problem With CC* (ETCC) ou Controle de Topologia em Comunicação Cooperativa Eficiente em Energia. Este problema consiste em obter uma topologia em que o consumo de energia total da rede é mínimo. Dada uma rede modelada como um grafo  $G(V, E)$  que é fortemente conectada sobre o modelo de comunicação cooperativa, o problema consiste em atribuir uma potência de transmissão individual  $P_i$  a cada nó tal que a topologia induzida é mantida dentro de um patamar de referência e a soma da potência de transmissão dos nós é minimizada. Os autores provam que o problema é NP-completo. São apresentadas duas abordagens gulosas para realizar o controle da topologia. As soluções mantêm o consumo de energia total dentro de um patamar estabelecido como parâmetro e consideram como referência os caminhos produzidos. O primeiro algoritmo, chamado *GreedyDelLink*, parte de uma topologia que utiliza comunicação cooperativa e, gradualmente, remove as arestas com menor eficiência energética tal que o patamar de energia definido seja respeitado. O segundo algoritmo, chamado *GreedyAddLink* adiciona, paulatinamente, enlaces mais eficientes até que o grafo resultante satisfaça o patamar de energia definido. Nestas soluções, as arestas mais eficientes são aquelas que, ao serem adicionadas, reduzem o fator de energia. Os autores argumentam que, apesar dos algoritmos serem centralizados, versões localizadas ou distribuídas podem ser facilmente desenvolvidas. Estas propostas não foram apresentadas. As soluções foram avaliadas por meio de simulações. Os algoritmos *GreedyDelLink* e *GreedyAddLink* são comparados com o *CoopBridges* [61]. Dados de simulação apontam que as soluções propostas mantêm um maior número de enlaces que o *CoopBridges*. Assim, o custo total dos enlaces criados é maior. Porém, demonstraram que as soluções propostas são capazes de manter o consumo de energia dentro dos patamares estabelecidos, na medida em que o *CoopBridges* não consegue obedecer o limite definido. Este comportamento se torna perceptível ao observar os caminhos de menor consumo de energia. Como o *CoopBridges* mantêm o único enlace de menor custo entre os componentes, as rotas devem convergir para este enlace. Em contrapartida, tanto o *GreedyAddLink* e o *GreedyDelLink* mantêm um maior número de arestas cooperativas. As arestas adicionais permitem que caminhos de menor custo sejam encontrados entre os componentes. Ao variar o número de nós entre 10 e 100 e estabelecer o limite para o fator da potência em 1,6, entre a topologia original e aquela com arestas cooperativas, as soluções propostas conseguiram atingir o objetivo, enquanto o *CoopBridges* apresentou

valores entre 1,4 e 11,0.

Neves e Bordim [36] utilizaram a comunicação cooperativa para aumentar a conectividade dos componentes de uma rede *ad hoc* com o nó sorvedouro. O algoritmo proposto, chamado *CoopSink*, liga os diversos componentes da rede ao sorvedouro utilizando Comunicação Cooperativa, enquanto reduz o consumo de energia das rotas. A solução é composta por quatro passos: (i) gerar uma topologia UDG utilizando a potência máxima de transmissão; (ii) criar todas as arestas cooperativas possíveis entre os nós da rede; (iii) realizar o CT considerando as rotas mais eficientes: as arestas de maior peso são removidas, exceto quando participam da primeira fase da comunicação cooperativa (enlaces auxiliares); e (iv) ajustar a potência de transmissão dos nós considerando o mínimo necessário para manter os enlaces diretos e os cooperativos. A topologia resultante, ao aplicar o *CoopSink*, é uma árvore. Para selecionar os auxiliares, este trabalho utiliza uma versão modificada do algoritmo GHSS [61] que otimiza o tempo de execução. A Figura 3.8 mostra um exemplo para o *CoopSink*. A Figura 3.8(a) apresenta o grafo inicial cuja potência de todos os nós é máxima. Nesta topologia, existem componentes conectados e nós isolados. A Figura 3.8(b) exhibe o grafo com todos os enlaces cooperativos que podem ser criados. A Figura 3.8(c) ilustra a topologia resultante após o CT sobre os enlaces diretos. A Figura 3.8(d) apresenta a topologia resultante após a união dos enlaces diretos de menor custo com os enlaces cooperativos de menor custo. Os nós que não possuem conectividade com o sorvedouro são ignorados na última etapa. Neste caso, contribuem negativamente para a métrica que estabelece o desempenho das soluções em aumentar a conectividade com o sorvedouro. A solução foi comparada ao *CoopSink* [61] em um cenário onde o nó sorvedouro foi inserido no centro de uma área de  $300m \times 300m$ . Os resultados demonstram que o *CoopSink* é capaz de aumentar a conectividade da rede com um pequeno aumento na potência individual dos nós. Contudo, a potência (e consequentemente a energia) necessária para atingir o sorvedouro é reduzida.

Rêgo *et al.* [45] propuseram uma alternativa para conectar os nós de uma rede de sensores ao sorvedouro, considerando o custo de energia e o atraso dos pacotes da rede. Esta publicação compõe as contribuições obtidas na pesquisa desenvolvida para esta tese. A solução considera que o nó sorvedouro detêm informações sobre a localização dos nós e não possui restrições de energia. A antena do sorvedouro é capaz de alcançar qualquer nó da rede, enquanto os sensores encaminham os dados para o sorvedouro por meio de caminhos com múltiplos saltos. A solução é executada em quatro passos: (i) construção de um grafo direcional utilizando a potência máxima em todos os nós; (ii) cálculo de todas as arestas cooperativas possíveis; (iii) definição, para cada nó, do próximo salto no caminho para o sorvedouro por meio de duas opções: utilizando o caminho com a menor quantidade de saltos para o sorvedouro ou o menor número de transmissões no caminho;

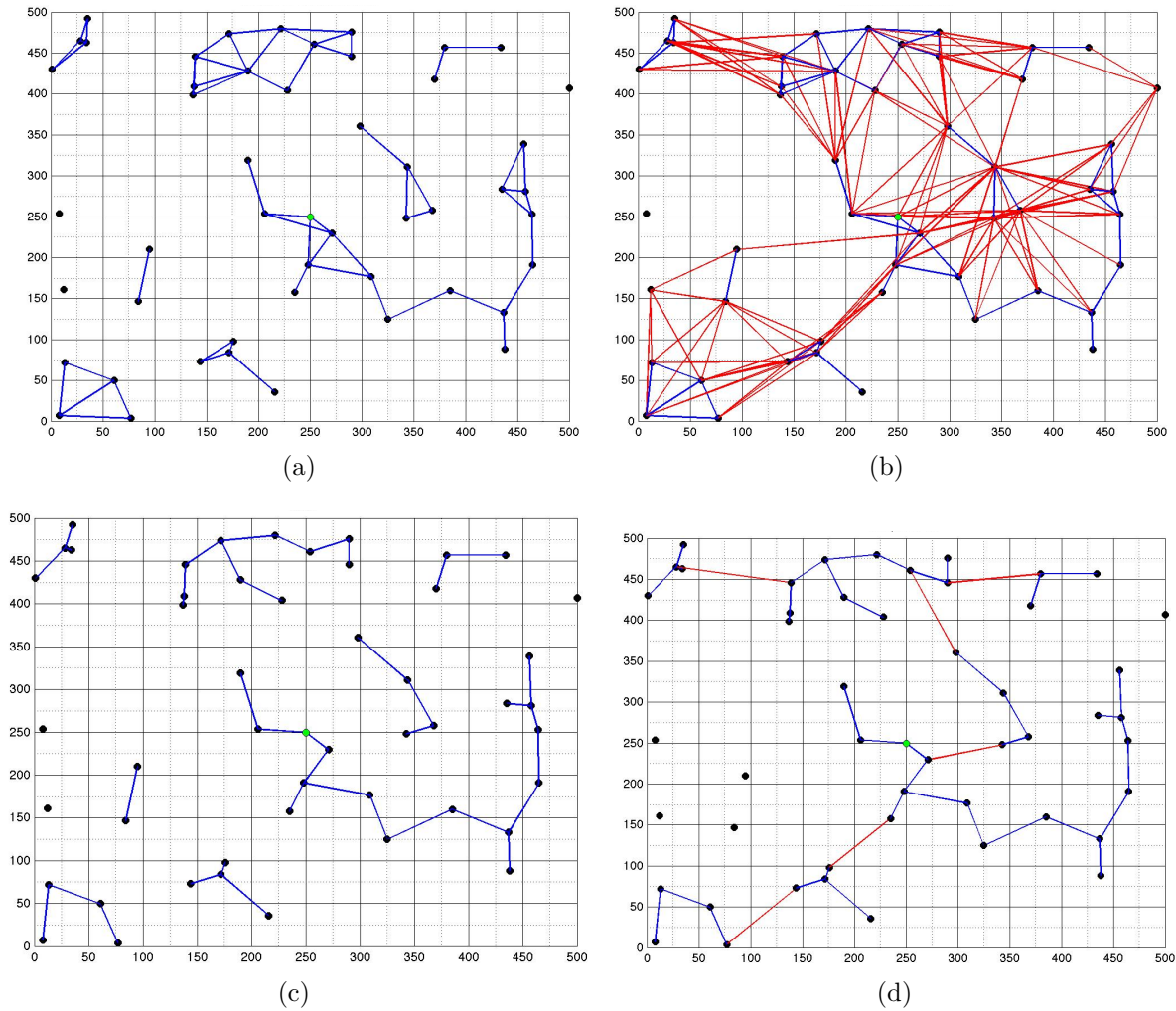


Figura 3.8: Exemplo do *CoopSink* (Fonte: [36]): (a) grafo inicial com a potência máxima de transmissão dos nós; (b) arestas cooperativas; (c) grafo resultante do Controle de Topologia sobre os enlaces diretos; (d) grafo resultante após a combinação dos enlaces diretos remanescentes e dos enlaces cooperativos de menor custo.

(iv) definição da potência de transmissão dos nós tal que os enlaces diretos e os cooperativos possam ser mantidos. Quando se utiliza o caminho com a menor quantidade de saltos para o sorvedouro, a proposta é chamada *CSC* (do Inglês *Cooperative Sink Communication*), enquanto a alternativa que seleciona o caminho com o menor montante de transmissões é chamada *CSC(HS)* (do Inglês *Cooperative Sink Communication Helper Set*). Para a seleção dos auxiliares, a heurística GHSS, de Yu *et al.* [61], foi aplicada. As soluções propostas conseguiram reduzir em 60% o número de saltos para o sorvedouro comparada ao *CoopSink* [61]. As soluções resultaram em um atraso de 54,6ms, à medida em que o *CoopSink* obteve um atraso de 109,2ms. As soluções propostas conseguiram reduzir a quantidade de saltos para o sorvedouro (e, por consequência, o número de transmissões necessárias) em relação ao *CoopSink*. Para os cenários avaliados, as soluções

conseguiram aumentar a conectividade da rede para 85%, enquanto apenas 35% da rede é conectada utilizando somente enlaces diretos. As soluções *CSC* e *CSC(HS)* apresentam um pequeno aumento na potência de transmissão individual dos nós em relação ao *CoopSink* como resultado do maior número de enlaces cooperativos mantidos. Contudo, a existência de uma maior quantidade de enlaces provê caminhos de menor custo para o sorvedouro.

Tabela 3.1: Controle de topologia baseado em Comunicação Cooperativa.

Algoritmo	Complexidade	Tipo	Saltos
DTCC [11]	$O( P(i)  \cdot (\Delta(G))^2)$	Localizado	2
ITCC [11]	$O((\Delta(G))^4)$	Localizado	1
CoopBridges [61]	$O( V ^2 +  V  \cdot \log  V  +  E  \cdot \log  E )$	Centralizado	-
GreedDelLink [62]	$O( E  \cdot  V  \cdot ( V  \cdot \log  V  +  E ))$	Centralizado	-
GreedAddLink [62]	$O( E  \cdot  V  \cdot ( V  \cdot \log  V  +  E ))$	Centralizado	-
CoopSink [36]	$O( V ^2)$	Centralizado	-
CSC e CSC(HS) [45]	$O( V ^2)$	Centralizado	-

A Tabela 3.1 apresenta um resumo dos algoritmos de Controle de Topologia baseados na Comunicação Cooperativa. Note que a complexidade dos algoritmos *CoopBridges*, *CoopSink*, *CSC* e *CSC(HS)* são computadas em função da utilização da rotina GHSS para selecionar os auxiliares. Os algoritmos DTCC e ITCC têm a complexidade definida em termos do grau do grafo  $\Delta(G)$  e dos intervalos para a potência de transmissão dos nós  $|P(i)|$ . Para o algoritmo *CSC* ainda é necessário definir o menor caminho para o sorvedouro, que pode ser realizado em  $O(|E| + |V| \log |V|)$ . Para os algoritmos *DTCC* e *ITCC*, não foram considerados os custos das técnicas de CT, que devem ser aplicadas sobre a topologia inicial antes mesmo da aplicação das soluções propostas. Note que algoritmos *DTCC* e *ITCC* tem como objetivo a redução no custo de energia da rede. Os algoritmos *CoopBridges*, *GreedDelLink*, *GreedAddLink* e *CoopSink* tem por objetivo incrementar a conectividade da rede criando enlaces entre componentes que, inicialmente, estão desconectados. Estes algoritmos, de fato não garantem a conectividade da rede. Assim como ilustrado nos exemplos das Figuras 3.7 e 3.8, a topologia inicial e final podem conter nós isolados. Os algoritmos apresentados nesta seção requerem informação global da topologia. As soluções são adequadas para o processo de implantação de rede *ad hoc* sem fio já que o custo dos algoritmos torna inviáveis utilizá-los em tempo de execução, sobretudo de forma recorrente.

### 3.4 Considerações

A Comunicação Cooperativa permite a criação de um sistema MIMO virtual em que nós portadores de uma única antena se coordenam para transmitir múltiplas versões de um mesmo sinal. A técnica combate o desvanecimento do sinal e, aplicada de forma adequada, permite aumentar a eficiência da rede em relação ao consumo de energia e vazão. As características de mitigação do desvanecimento de sinal permitem que nós extrapolem o alcance máximo. Assim, a CC foi utilizada como mecanismo de Controle de Topologia para incrementar o número de arestas da rede.

No contexto do Controle de Topologia, os enlaces cooperativos foram utilizados para incrementar a conectividade da rede, em especial, para conectar conjuntos de nós que inicialmente estão isolados. Estas soluções têm como requisito a informação global da topologia e são centralizadas em relação ao processamento. Elas requerem como entrada um grafo com múltiplos componentes de rede e produzem uma topologia que combinam enlaces cooperativos e diretos para garantir a existência de caminhos entre o maior número possível de nós. A necessidade de coletar informações globais da topologia da rede e alto custo computacional tornam estas soluções inviáveis, sobretudo durante a operação da rede. Assim, não são passíveis para cenários de recuperação de conectividade. Porém, a possibilidade de usar a Comunicação Cooperativa para extrapolar o alcance máximo dos nós pode ser utilizado como mecanismo de recuperação de conectividade. Considerando o estudo realizado, podemos elencar as seguintes alternativas para aplicação da Comunicação Cooperativa:

- utilização de enlaces cooperativos para recuperar a conectividade por meio da previsão de falhas em elementos críticos;
- utilização da notificação de erros em algoritmos de roteamento em redes *ad hoc* sem fio (como, por exemplo, no *Ad Hoc On-Demand Distance Vector* [41]) para detectar falhas na conectividade e utilizar a Comunicação Cooperativa para alcançar outros componentes de rede;
- verificar as consequências do efeito de borda na Comunicação Cooperativa, considerando que os nós com maior distância em relação ao centro da rede possuem, potencialmente, um menor número de candidatos e auxiliares e, portanto, menor possibilidade de dispor dos benefícios da diversidade espacial distribuída;

Os Capítulos 4 e 5, a seguir, apresentam as soluções propostas nesta tese. Inicialmente são apresentados algoritmos distribuídos para a localização de articulações, pontes e componentes biconectados. Estas informações são utilizadas para prever os pontos de segmentação e os componentes formados pelas falhas em elementos críticos. Em seguida,

são apresentadas duas soluções distribuídas, baseadas na Comunicação Cooperativa, que permitem a recuperação de conectividade da rede por meio enlaces cooperativos criados com informações localizadas da topologia.

## Capítulo 4

# Localização de Elementos Críticos e Componentes Biconectados

Este capítulo apresenta um conjunto de algoritmos para localizar articulações, pontes e componentes biconectados em uma rede *ad hoc* sem fio. Os algoritmos são interdependentes e a aplicação deve ser realizada sequencialmente na ordem em que são apresentados. Porém, o uso de todos os algoritmos não é obrigatório. A solução permite determinar: (a) somente as articulações; (b) as articulações e pontes; ou (c) as articulações, pontes e os componentes biconectados. Os algoritmos podem ser utilizados na inicialização da rede para obter informações sobre os pontos de falha. Apesar de não serem projetados especificamente para redes móveis, o baixo custo permite a utilização de forma periódica para acomodar mudanças na topologia.

Presume-se que o grafo de entrada  $G(V, E)$ , é conectado, possui um número de vértices  $|V| \geq 3$  e corresponde ao UDG com a potência máxima de transmissão para todos os nós. Os algoritmos são distribuídos e requerem apenas o conhecimento dos nós diretamente conectados. Ao final da execução, cada nó detém um conjunto de variáveis que indicam: (i) se são articulações; (ii) se participam de pontes; (iii) quais os nós vizinhos que compartilham as pontes; e (iv) um identificador para o componente biconectado a que pertence. A interação entre os algoritmos ocorre por meio de variáveis armazenadas em memória. Ao final da execução do primeiro algoritmo, cada nó possuirá duas variáveis acessíveis ao segundo algoritmo: uma para armazenar a informação se o nó é uma articulação e outra para lista de arestas que não fazem parte da árvore distribuída. Ao final da execução do segundo algoritmo, as seguintes informações são armazenadas em memória: uma variável que define se o nó participa de pontes e outra que armazena a lista de nós que compõe as pontes. O resto deste capítulo é organizado como se segue. A Seção 4.1 apresenta uma solução distribuída baseada no DDFS para a localização de articulações. A Seção 4.2 detalha a solução distribuída para a localização de pontes. A Seção 4.3 explica



a solução para a localização de componentes biconectados. Os resultados numéricos são apresentados na Seção 4.4. Estes resultados têm como objetivo a definição de parâmetros de simulação para algoritmos de recuperação de conectividade. A Seção 4.5 resume as principais contribuições para as soluções apresentadas neste capítulo.

## 4.1 Algoritmo Distribuído Para Localização de Articulações

Esta seção apresenta uma solução distribuída, chamada *Distributed Articulation Search* (DAS) ou Busca Distribuída por Articulações, para localizar articulações em redes *ad hoc* sem fio. A solução é baseada no *Distributed Depth First Search* (DDFS), ou Algoritmo de Busca em Profundidade Distribuído e nas premissas elaboradas por Hopcroft e Tarjan [22] para determinar as articulações e pontes de um grafo. O objetivo é reduzir, em relação à proposta de Chaudhuri [14], o custo computacional em termos de comunicação e o número de tipos diferentes de mensagens. A redução no volume de dados necessários, para realizar esta tarefa, se dá pela redução na quantidade de mensagens e pela redução na quantidade de informações de controle.

O DAS tem como entrada um grafo conectado  $G(V, E)$  e um parâmetro *root* que indica o nó utilizado como raiz da árvore distribuída. A raiz da árvore pode ser escolhida de forma arbitrária. Cada nó deve possuir um identificador único e conhecer seus vizinhos diretamente conectados. O algoritmo cria uma árvore distribuída utilizando uma mensagem especial *SEARCH* que carrega: (i) uma lista de nós ancestrais  $a$ ; (ii) uma lista  $v$  de todos os nós visitados até o momento; e (iii) uma lista  $nt$  de arestas *nontree*, ou arestas que não pertencem à árvore. A mensagem pode ser enviada em duas direções: para os filhos ou para o nó pai. A mensagem *SEARCH* enviada para os filhos é utilizada para adicionar nós ainda não visitados à árvore e representar esta informação em memória. Quando é enviada para o nó pai, determina que uma subárvore foi completamente pesquisada e propaga a lista de arestas que não pertencem à árvore, encontradas pelo próprio nó e por seus descendentes. Ao receber uma mensagem *SEARCH* de um filho, os nós podem determinar se são articulações. O algoritmo encerra quando a raiz recebe uma mensagem de retorno de todos os filhos. Ao final da execução, cada nó sabe, por meio de uma variável local, se é uma articulação. O Algoritmo 2 apresenta os detalhes do DAS. Em seguida, o algoritmo é detalhado e exemplificado.

Durante a inicialização dos nós, são criadas variáveis para armazenar a identificação do nó pai, do conjunto de nós filhos, do conjunto de ancestrais, do conjunto de arestas que não pertencem à árvore, de nós já visitados e, daqueles que ainda não foram visitados (linhas 1-4). Quando o nó é a raiz, um vizinho é escolhido, adicionado à lista de filhos e de

---

**Algorithm 2** DistributedArticulationSearch (root)

---

**Require:** Each node  $v_i$  must know its direct neighbors  $N[v_i]$ .

**Ensure:** Each node knows, using its logical local variable  $articulation_{v_i}$ , if it is an articulation node.

# Initialization of node  $v_i$

```
1:  $parent_{v_i} \leftarrow 0$ ;  
2:  $children_{v_i} \leftarrow ancestor_{v_i} \leftarrow nontree_{v_i} \leftarrow \emptyset$ ;  
3:  $visited_{v_i} \leftarrow \{v_i\}$ ;  
4:  $tovisit_{v_i} \leftarrow N[v_i]$ ;  
5: if  $v_i = root$  then  
6:    $destination_{v_i} \leftarrow$  first node in  $tovisit_{v_i}$ ;  
7:    $children_{v_i} \leftarrow children_{v_i} \cup \{destination_{v_i}\}$ ;  
8:    $tovisit_{v_i} \leftarrow tovisit_{v_i} \setminus \{destination_{v_i}\}$ ;  
9:    $visited_{v_i} \leftarrow visited_{v_i} \cup \{destination_{v_i}\}$ ;  
10:  send message  $SEARCH(\{v_i\}, \{v_i\}, \emptyset)$  to  $destination$ ;  
11: end if
```

# Node  $v_j$  receive a  $SEARCH(a, v, nt)$  message from  $v_i$

```
12:  $visited_{v_j} \leftarrow visited_{v_j} \cup v$ ;  
13:  $tovisit_{v_j} \leftarrow tovisit_{v_j} - visited_{v_j}$ ;  
14: if  $parent_{v_j} = 0$  and  $v_j \neq root$  then  
15:    $parent_{v_j} \leftarrow v_i$ ;  
16:    $ancestor_{v_j} \leftarrow ancestor_{v_j} \cup a$ ;  
17: end if  
18: if  $v_i \in children_{v_j}$  then  
19:   if  $v_j = root$  then  
20:     if  $|children| > 1$  then  
21:        $articulation_{v_j} \leftarrow TRUE$ ;  
22:     end if  
23:   else  
24:     if  $\nexists (v_x v_y) \in nt \mid x \in ancestor_{v_j} \text{ or } y \in ancestor_{v_j}$  then  
25:        $articulation_{v_j} \leftarrow TRUE$ ;  
26:     else  
27:        $dualLink_{v_j} \leftarrow v_i$ ;  
28:     end if  
29:      $nontree_{v_j} \leftarrow nontree_{v_j} \cup nt$ ;  
30:   end if  
31: end if  
32: if  $tovisit_{v_j} \neq \emptyset$  then  
33:    $destination \leftarrow$  first node in  $tovisit_{v_j}$ ;  
34:    $tovisit_{v_i} \leftarrow tovisit_{v_i} - destination_{v_i}$ ;  
35:    $children_{v_j} \leftarrow children_{v_j} \cup destination_{v_j}$ ;  
36:   send message  $SEARCH(ancestor_{v_j} \cup \{j\}, visited_{v_j}, \emptyset)$  to  $destination_{v_j}$ ;  
37: else  
38:   if  $v_j \neq root$  then  
39:     for all  $v_k \in \{neighbour_{v_j} \setminus (children_{v_j} \cup \{parent_{v_j}\})\}$  do  
40:        $nontree_{v_j} \leftarrow nontree_{v_j} \cup \{v_j v_k\}$ ;  
41:     end for  
42:     send  $SEARCH(\emptyset, visited_{v_j}, nontree_{v_j})$  to node  $parent_{v_j}$ ;  
43:   end if  
44: end if
```

---

nós já visitados e a primeira mensagem da busca em profundidade é enviada (linhas 5-11). Mensagens  $SEARCH$  recebidas pelos nós são o gatilho para a execução do restante do algoritmo.

Quando um nó recebe uma mensagem, ele atualiza a lista de nós já visitados e a visitar por meio do conteúdo da mensagem (linhas 12-13). Se o nó não é a raiz da árvore e, ainda não foi visitado ( $parent_{v_j} = 0$ ), então o nó que enviou a mensagem é atribuído como nó pai e a lista dos ancestrais é atualizada por meio do conteúdo da mensagem recebida (linhas 15-17). Se a mensagem foi enviada por um nó filho, as linhas 18-31 são executadas. Neste caso, se o nó que recebeu a mensagem é a articulação, ela determina que é uma articulação se possuir mais de um filho (linhas 19-23). Se o nó que recebe a mensagem é um nó ordinário, ele verifica se é uma articulação por meio da lista  $nt$  de arestas *nontree* (recebida no conteúdo da mensagem) e da lista de ancestrais (linhas 24-28). Em seguida, atualiza a lista de arestas que não pertencem à árvore, aplicando o conteúdo da mensagem (linha 29). Após o processamento das linhas 12-31, os nós devem continuar a construção da árvore distribuída. Se possuem nós ainda não visitados, um deles é retirado da lista, adicionado ao conjunto de filhos e, então uma mensagem *SEARCH* é enviada a este nó (linhas 32-36). Se o nó não possui vizinho para adicionar à árvore, ele constrói sua lista de arestas *nontree* e envia para o nó pai (linhas 37-43).

A Figura 4.1 ilustra a execução do algoritmo DAS em um grafo com seis nós. O nó  $v_1$  é utilizado como raiz da árvore, isto é,  $root = v_1$ . A Figura 4.1(a) apresenta a troca das três primeiras mensagens. A busca é realizada até que uma folha é encontrada (não existem vizinhos para adicionar à árvore). As arestas  $v_4v_1$  e  $v_4v_2$  são marcadas como *nontree*. Na Figura 4.1(b), mensagens de retorno são enviadas até que seja encontrado um nó que ainda possui vizinhos não adicionados à árvore. O nó  $v_2$  não é marcado como articulação, já que a origem da mensagem possui arestas para um de seus ancestrais. O processo de inclusão de nós à árvore continua na Figura 4.1(c) até a folha  $v_5$ . Em seguida, conforme ilustrado na Figura 4.1(d), o nó  $v_5$  determina que a aresta  $v_5v_2$  é *nontree* e propaga esta informação até  $v_6$ . O nó  $v_2$  identifica que é uma articulação já que recebe uma mensagem do nó filho  $v_6$  com uma lista de arestas *nontree* que não inclui um de seus ancestrais. Durante a propagação de mensagens para o nó pai (mensagem 4, 5, 8, 9, 10), a lista de arestas *nontree* recebida é adicionada ao conhecimento local do nó e propagado para os ancestrais. Quando o nó  $v_1$  recebe a mensagem 10, a busca encerra, já que ele é a raiz e não existem nós para adicionar à árvore.

Durante a construção da árvore, podem existir duas arestas a partir da articulação, que levam a um mesmo componente biconectado. Isto ocorre quando uma articulação é encontrada e algum nó daquele componente ainda não foi visitado. Para facilitar o processo de definição dos componentes biconectados, o algoritmo mantém (na articulação) uma variável chamada *dualLink*, contendo a identificação do nó filho que compõe a aresta. A linha 27 do algoritmo armazena esta informação quando o nó é uma articulação e recebe uma mensagem *SEARCH* de um filho, mas para aquela subárvore, especificamente, o

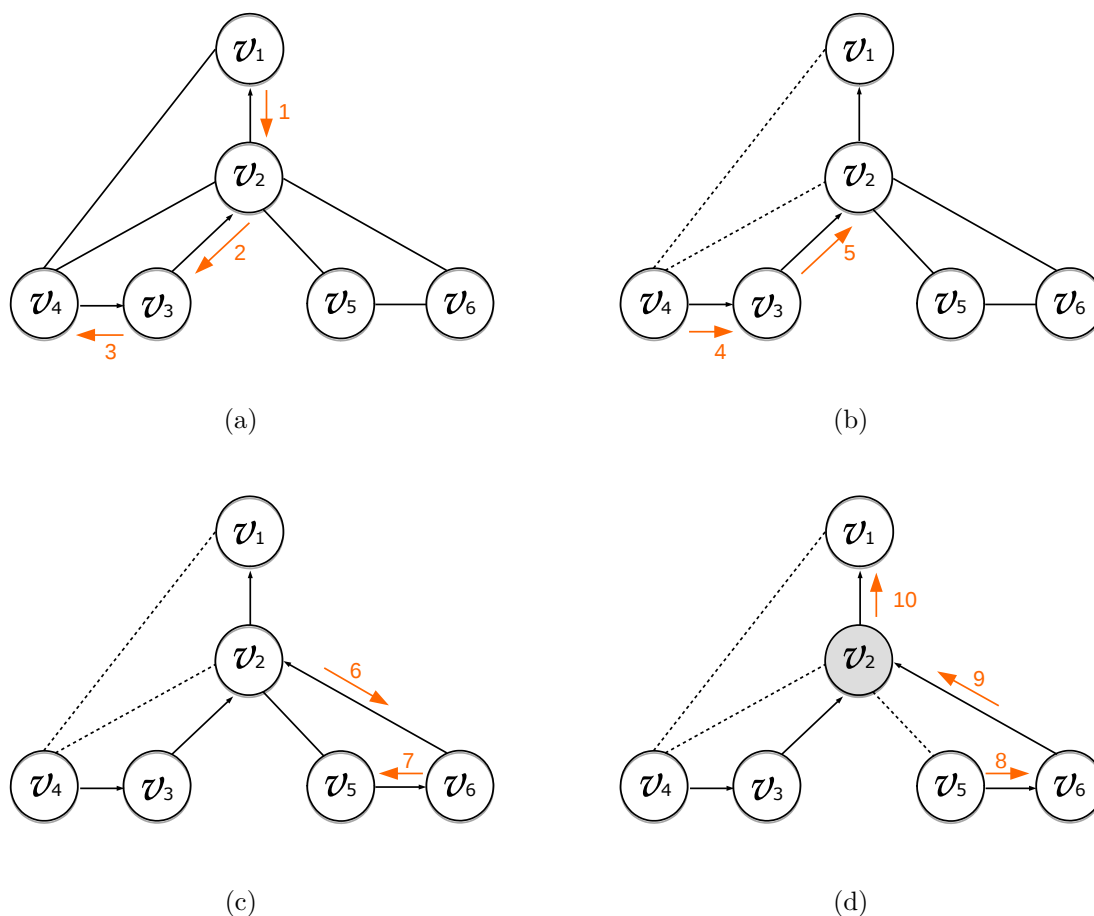


Figura 4.1: Exemplo do algoritmo DAS.

nó não constitui uma articulação (condição das linha 25 e 27). Em outras palavras, a articulação recebeu a mensagem de um filho que possui uma aresta *nontree* para um de seus ancestrais.

Para fins de comparação, a complexidade do algoritmo DAS pode ser calculada em relação ao número de mensagens. Dado um grafo  $G(V, E)$ , o algoritmo cria uma árvore  $T(V, E')$  distribuída. Pelas propriedades apresentadas no Capítulo 2, existem  $|V| - 1$  arestas em  $T$ . Pode-se notar que, para cada aresta da árvore resultante, são transferidas exatamente duas mensagens *SEARCH*, uma de busca e outra de retorno. A mensagem de busca é enviada somente quando um nó ainda não foi visitado, propriedade garantida pelas linhas 8, 14 e 35. Já a segunda, de retorno, é enviada apenas para o nó pai (conforme linha, 43). Dessa forma, apenas as arestas que fazem parte da árvore são utilizadas para transportar a mensagem de busca e retorno de cada nó. Como um nó não é incluído duas vezes na árvore, são utilizadas, no total,  $2 \cdot |E'|$  mensagens. Em relação à quantidade de nós, são enviadas  $2 \cdot (V - 1)$  mensagens. Assim, para o grafo  $G(V, E)$ , a complexidade do

algoritmo DAS é  $O(|V|)$ . O Lema 4.1.1 resume a discussão acima.

**Lema 4.1.1.** *Dado uma topologia de rede representada por um grafo  $G(V, E)$ , com  $|V|$  nós e  $|E|$  arestas, o algoritmo DAS utiliza no máximo  $2 \times |V|$  mensagens.*

*Demonstração do Lema 4.1.1.* Considere um grafo  $G(V, E)$  qualquer. Considere também que, durante a construção de  $T(V, E')$ ,  $\text{parent}(v_j) = v_i$ , isto é,  $v_i$  adicionou o nó  $v_j$  à lista de nós filhos, enquanto  $v_j$  recebeu uma mensagem de busca de  $v_i$  e o atribuiu como nó pai. Será provado que, para cada nó  $v_i$  apenas uma mensagem de busca e outra de retorno são transmitidas em um enlace  $v_i v_j \in E'$  durante a execução do algoritmo DAS. Para tal, assumamos que  $v_j$  recebeu outra mensagem de busca a partir de  $v_k$ . Isso é uma contradição, pois o DAS realiza uma busca em profundidade, onde cada nó pai envia apenas uma mensagem para cada nó filho. A adição do enlace  $v_j v_k$  produz um ciclo. Portanto, exatamente uma mensagem de busca pode ser enviada por meio de um enlace em  $E'$ . Agora, assumamos que o algoritmo terminou e  $v_j$  não enviou uma mensagem de retorno para  $v_k$ . Trata-se de uma contradição, já que, se  $v_j$  não enviou a mensagem de retorno, então ainda existem nós a visitar a partir dele. Assumamos que  $v_i$  recebeu duas mensagens de retorno a partir de  $v_j$ . Trata-se de uma contradição, pois os nós só enviam mensagens de retorno quando visitam todos os nós a partir dele. Se ele enviou a primeira mensagem é porque todos os nós foram visitados e a segunda mensagem não é possível. Portanto, apenas uma mensagem de retorno pode ser transmitida a partir de  $v_i v_j$ . Assim, a partir do grafo  $G(V, E)$ , uma árvore  $T(V, E')$  é criada com o algoritmo DAS, enquanto exatamente duas mensagens são enviadas por meio de cada enlace em  $E'$ . Como  $|E'| = |V| - 1$ , então a complexidade do DAS em termos de mensagens é  $O(|V|)$ .  $\square$

O DAS mantém as propriedades do algoritmo DDFS e as premissas elaboradas em [22] para localizar as articulações. Além disso, mantém as mesmas estruturas propostas por Chaudhuri [14]. Desta forma, a corretude e completude do DAS podem ser provadas de forma similar. As provas podem ser derivadas diretamente do próprio DDFS, assim como realizado em [14] e [56]. Outra propriedade interessante do DAS é a possibilidade de reduzir a quantidade de dados enviados pelos nós durante a busca. Ao enviar mensagens  $SEARCH(a, v, nt)$ , os nós ancestrais foram previamente visitados, então  $a \subseteq v$ . O algoritmo pode enviar apenas a lista dos ancestrais no campo  $a$  da mensagem, enquanto o campo  $v$  pode incluir apenas a interseção entre os nós já visitados e os ancestrais ( $v \cap a$ ). De fato, o algoritmo pode reconstruir a lista de nós visitados com os campos  $a$  e  $v$ . Desta forma, cada mensagem carregará, no pior caso,  $|V|$  identificadores de nós.

## 4.2 Algoritmo Distribuído Para Localização de Pontes

Esta seção apresenta um mecanismo distribuído para localizar pontes, em um grafo conectado  $G(V, E)$ . A solução, chamada *Distributed Bridge Search* (DBS) ou Busca Distribuída por Pontes, tem como requisito as variáveis locais produzidas pelo algoritmo DAS e como parâmetro a raiz da árvore distribuída. Isto é, o algoritmo requer o conhecimento da árvore  $T(V, E')$ ,  $E' \subset E$ , representada de forma distribuída na memória dos nós, assim como a identificação do nó utilizado como raiz durante sua construção. A ideia geral é marcar como pontes os enlaces que: (i) ligam um nó folha da árvore a seu ancestral, quando este não possuir uma aresta *nontree* (que não pertence à árvore); e (ii) ligam uma articulação a um nó ancestral que também é articulação. O algoritmo é executado em todos os nós da rede e o gatilho para execução é a definição de um nó como articulação ou como folha. Assim, a troca de mensagens é realizada apenas pelos nós que podem, efetivamente, constituir pontes. Uma mensagem especial *BRIDGE* é enviada pelas articulações e por nós que são folhas na árvore  $T(V, E')$  não possuem arestas *nontree* no grafo  $G(V, E)$ . Ao receber a mensagem, os nós enviam uma mensagem *BRIDGE* de resposta quando verificam que o enlace realmente constitui uma ponte. Os detalhes deste mecanismo são apresentados no Algoritmo 3. Em seguida o algoritmo é explicado e exemplificado.

Durante a inicialização, são criadas variáveis para determinar se o nó participa de uma ponte e para armazenar a lista de nós que compõem as pontes incidentes (linhas 1-2). Então, o nó verifica se é uma folha e se a lista de arestas *nontree* é vazia. Neste caso, envia uma mensagem *SEARCH* para seu ancestral indicando que compõe uma ponte (linhas 3-5). Se o nó é uma articulação e não é a raiz, ele pode possuir pontes (condição da linha 6). Ele verifica se possui uma aresta que liga a um ancestral e, se não existir, envia uma mensagem *BRIDGE* para seu ancestral indicando que a aresta pode constituir uma ponte (linhas 6-10).

Após a inicialização, o restante do algoritmo é executado apenas quando uma mensagem *BRIDGE* é recebida pelo nó (linhas 11-20). Sempre que o nó recebe uma mensagem, deve verificar se compõe uma ponte com a origem e, neste caso, confirmar com outra mensagem *SEARCH*. Se o nó recebe a mensagem do nó pai, isto indica que é uma confirmação e então o nó marca a existência da ponte e adiciona a origem à lista de pares que compõem pontes (linhas 11-13). Quando o nó recebe a mensagem de um filho e o próprio nó é uma articulação, então a aresta é uma ponte e a informação é adicionada às variáveis de memória (linhas 16-20).

A Figura 4.2 exemplifica o algoritmo DBS. A Figura 4.2(a), apresenta a entrada para o algoritmo: um grafo com as articulações identificadas e uma árvore representada na

---

**Algorithm 3** DistributedBridgeSearch (root)

---

**Require:** DAS internal structures  $parent_{v_i}$ ,  $children_{v_i}$ ,  $nontree_{v_i}$ ,  $articulation_{v_i}$  and  $ancestor_{v_i}$ ;

**Ensure:** Each node knows, using its logical local variable  $bridge_{v_i}$ , if it participates in a bridge and, the list of nodes participating in the bridges using the local array  $bridgePair_{v_i}$ .

```
# Initialization of node  $v_i$ 
1:  $bridge_{v_i} \leftarrow FALSE$ ;
2:  $bridgePair_{v_i} \leftarrow \emptyset$ ;
3: if  $children_{v_i} = \emptyset$  and  $nontree_{v_i} = \emptyset$  then
4:   send message  $BRIDGE(v_i)$  to node  $parent_{v_i}$ ;
5: end if
6: if  $articulation_{v_i} = TRUE$  and  $v_i \neq root$  then
7:   if  $\nexists v_x v_y \in nt \mid x \in ancestor_{v_i}$  or  $y \in ancestor_{v_i}$  then
8:     send message  $BRIDGE(v_i)$  to node  $parent_{v_i}$ ;
9:   end if
10: end if
# Node  $v_j$  receive a  $BRIDGE(v_i)$  message from node  $v_i$ 
11: if  $v_i = parent_{v_j}$  then
12:    $bridge_{v_j} \leftarrow TRUE$ ;
13:    $bridgePair_{v_j} \leftarrow bridgePair_{v_j} \cup \{v_i\}$ ;
14: else
15:   if ( $v_i \in children_{v_j}$  and  $articulation_{v_j} = TRUE$ ) or  $v_j = root$  then
16:      $bridge_{v_j} \leftarrow TRUE$ ;
17:      $bridgePair_{v_j} \leftarrow bridgePair_{v_j} \cup \{v_i\}$ ;
18:     send message  $BRIDGE(v_j)$  to node  $v_i$ ;
19:   end if
20: end if
```

---

memória dos nós de forma distribuída. As arestas com seta indicam o pai de cada nó e, as arestas pontilhadas representam arestas *nontree*. A Figura 4.2(b) representa a inicialização. A articulação  $v_4$  envia  $BRIDGE(4)$  para o nó pai  $v_3$ , indicando que aquela aresta pode constituir uma ponte. As articulações  $v_3$  e  $v_5$  não enviam mensagens já que não podem constituir pontes (condição da linha 7 do algoritmo). O nó folha  $v_7$  não possui arestas *nontree* e, portanto, envia  $BRIDGE(7)$  para o nó pai  $v_5$ . Na Figura 4.2(b), quando as articulações  $v_3$  e  $v_5$  recebem a mensagem, identificam que possuem uma ponte e que os nós  $v_4$  e  $7$ , respectivamente, são os nós que compartilham a ponte. Ao lado dos nós, estão representadas as variáveis locais utilizadas para armazenar tais informações. Note que o nó  $v_1$  ignora a mensagem já que não é uma articulação e, sendo assim, não compõe uma ponte. Na Figura 4.2(c), os nós que identificaram a existência de pontes enviam as mensagens  $BRIDGE(3)$  e  $BRIDGE(5)$  aos nós filhos. Quando os nós  $v_4$  e  $v_7$  recebem a mensagem confirmando a existência da ponte, inserem a informação na memória local. A Figura 4.2(d), apresenta o fim da execução do algoritmo. As pontes estão destacadas por meio de linhas espessas e, ao lado de cada nó, está representado o conteúdo da memória para o algoritmo DBS.

A complexidade deste algoritmo pode ser computada em relação à quantidade de mensagens enviadas e em relação ao tempo de execução. Considere um grafo  $G(V, E)$

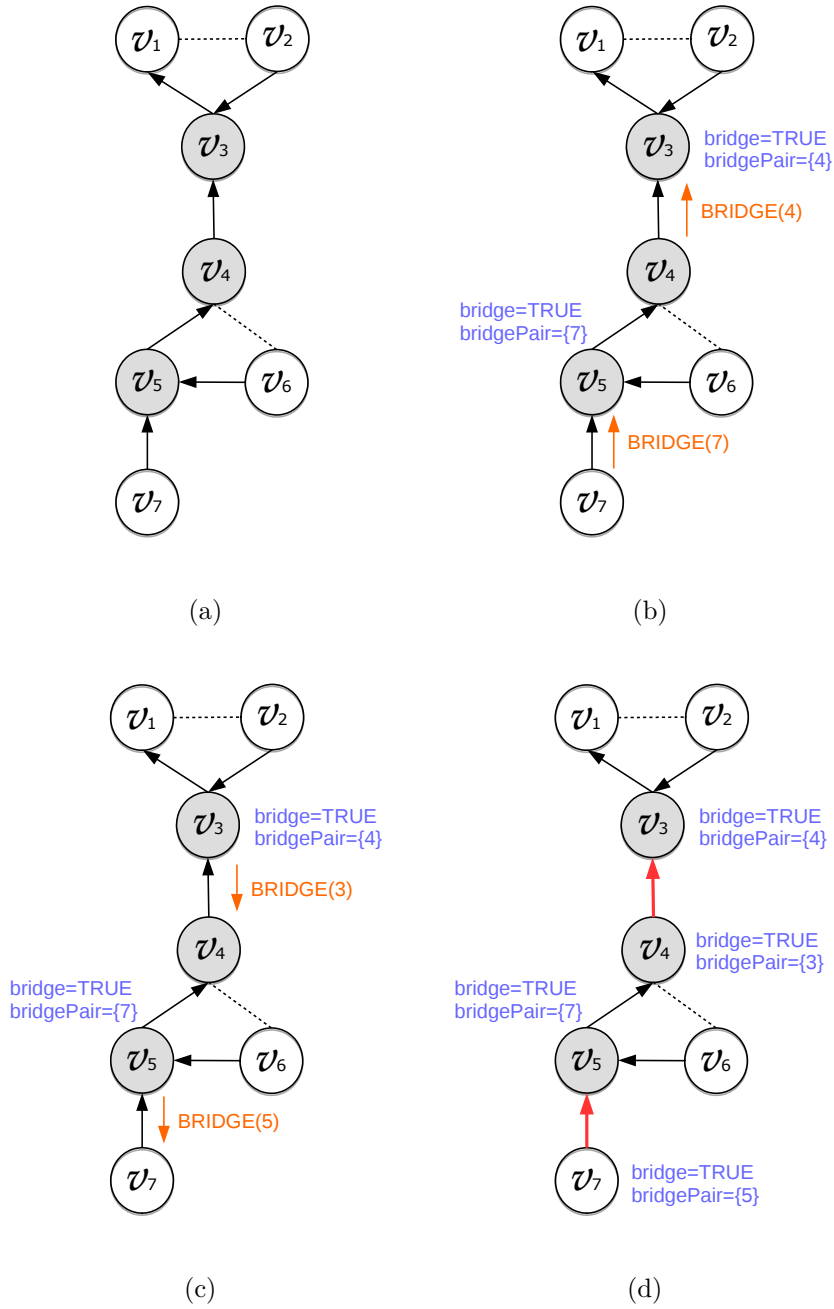


Figura 4.2: Exemplo do algoritmo DBS.

com um total de  $n_a$  articulações e  $n_f$  folhas e uma árvore  $T(V, E')$  produzida com o algoritmo DAS a partir de  $G(V, E)$ . No pior caso, todas as arestas do grafo  $G$  são pontes e o algoritmo DBS envia  $2 \times E$  mensagens. Para cada enlace, uma mensagem é enviada durante a inicialização, conforme as condições das linha 3 e 6 do algoritmo. A outra mensagem é enviada como confirmação da existência da uma ponte, por meio da condição da linha 16. Assim, o algoritmo determina as pontes do grafo  $G(V, E)$  por meio das arestas  $E'$  (produzidas pelo DAS) utilizando, no máximo,  $O(|V|)$  mensagens. O Lema



4.2.1 resume a discussão acima.

**Lema 4.2.1.** *Considere um grafo  $G(V, E)$  e uma árvore  $T(V, E')$ ,  $E' \subset E$  criada com o algoritmo DAS. O algoritmo DBS requer  $O(|V|)$  mensagens, no pior caso, para detectar as pontes.*

*Demonstração do Lema 4.2.1.* Considere uma árvore  $T(V, E')$  construída sobre o grafo  $G(V, E)$ , tal que  $E' \subset E$ , utilizando o algoritmo DAS. De acordo com o Lema 4.1.1, cada nó sabe se é uma articulação. O algoritmo ainda permite que cada nó saiba, por meio de variáveis locais, a identificação do nó pai e dos nós filhos na árvore. Dado um nó  $v_i$  qualquer, existem duas possibilidades: o nó  $v_i$  é uma articulação ou não. Suponha que  $v_i$  é um nó ordinário e que este enviou uma mensagem para  $v_j$ . Neste caso, se  $v_i$  não é uma folha da árvore, temos uma contradição, já que as linhas 3 e 6 garantem que as mensagens são enviadas apenas por articulações e nós folhas da árvore  $T$  que não possuem arestas *nontree* no grafo  $G$ . Então,  $v_i$  só pode ser uma articulação ou uma folha. Ao receber a mensagem,  $v_j$  pode ou não enviar uma mensagem de resposta para  $v_i$ . Suponha que  $v_j$  enviou uma mensagem para o nó  $v_i$  e que  $v_j$  é um nó ordinário. Temos outra contradição, já que a linha 15 garante que mensagens de resposta são enviadas apenas por articulações. Assim, mensagens são enviadas, durante a iniciação, pelas articulações e pelos nós folhas que não possuem arestas *nontree*, e como resposta, apenas pelas articulações que compõem pontes. São necessárias exatamente duas mensagens enviadas, para cada ponte, para determinar sua condição. Para um grafo  $G$  em que existem  $n_a$  articulações e  $n_f$  pontes, o algoritmo utiliza  $2 \times n_f$  mensagens para identificar as pontes e  $n_a - 2 \times n_f$  mensagens para identificar as arestas que não são pontes (quando a aresta liga uma articulação a um nó ordinário). No pior caso, todas as arestas são pontes. Neste caso, são utilizadas  $2 \times n_f = 2 \times |E'|$ . O custo de comunicação do DBS é  $O(|V|)$ .  $\square$

A Figura 4.3 apresenta um exemplo, para o algoritmo DBS, que ilustra o pior caso em relação à comunicação. Neste exemplo, não existem arestas *nontree*, isto é, todas as arestas do grafo fazem parte da árvore produzida com o DAS. Assim, todas as arestas são pontes. As arestas são representadas com setas que indicam a direção do nó pai. A raiz da árvore é o nó  $v_3$ , enquanto os nós  $v_1$  e  $v_5$  são folhas. Neste caso, os nós  $v_2$ ,  $v_3$  e  $v_4$  são articulações. Duas mensagens são enviadas por meio de cada aresta, representadas logo abaixo de cada enlace. A primeira seta, de cima para baixo, corresponde à mensagem *BRIDGE* enviada para verificar se o enlace é uma ponte e, a segunda, refere-se à confirmação.

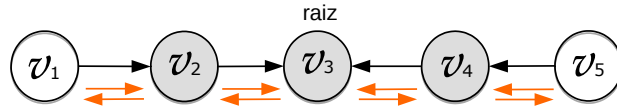


Figura 4.3: Exemplo de pior caso para o algoritmo DBS.

### 4.3 Algoritmo Distribuído Para Definição de Componentes Biconectados

Esta seção apresenta um mecanismo distribuído para determinar os componentes biconectados de um grafo. A solução, chamada *Distributed Define Biconnected Components* (DDBC) ou Busca Distribuída por Componentes Biconectados, tem como requisito as variáveis locais produzidas pelos algoritmos DAS e DBS e como parâmetro a raiz da árvore distribuída. Isto é, cada nó conhece o identificador do nó pai e os identificadores dos nós filhos na árvore distribuída  $T(V, E')$ ,  $E' \subset E$ , e tem o conhecimento se são articulações (resultado do algoritmo DAS). A raiz da árvore distribuída, fornecida como parâmetro, é utilizada para iniciar a definição dos componentes biconectados. Além disso, todos os nós sabem se existem pontes incidentes e quais vizinhos compõem estas pontes (resultado do algoritmo DBS). Como saída, cada nó detém uma variável local que armazena os identificadores dos componentes biconectados a que pertence. O identificador dos componentes biconectados utiliza valores inteiros não sequenciais que correspondem ao identificador do primeiro nó adicionado ao componente. Nós em componentes biconectados disjuntos possuem identificadores diferentes. A definição dos componentes biconectados pode auxiliar na previsão da topologia resultante quando uma articulação é removida. Neste trabalho, os componentes biconectados são utilizados para definir os pares de nós, um em cada componente, responsáveis por recuperar a conectividade.

O DDBC é baseado em uma mensagem especial chamada *COMPONENT*. Esta mensagem propaga um identificador de componente por todos os nós da rede utilizando a estrutura da árvore distribuída. A primeira mensagem é enviada pela raiz da árvore e as mensagens seguintes são enviadas de um nó pai aos nós filhos. A ideia geral da solução é propagar o identificador de componente biconectado do nó para seus filhos e este valor deve mudar quando uma articulação é encontrada. A partir daquele ponto, todos os nós devem possuir identificadores diferentes. O Algoritmo 4 apresenta com detalhes a solução proposta. Em seguida, o algoritmo é explicado e exemplificado.

Durante a inicialização, uma variável  $component_{v_i}$  é criada para armazenar os componentes biconectados a que o nó pertence (linha 1). Em seguida, a raiz inicia o processo de troca de mensagens por meio da árvore (condição da linha 2). Se a raiz é uma articulação, então possui mais de um filho [22]. Neste caso, a raiz adiciona o identificador

---

**Algorithm 4** DistributedDefineBiconnectedComponent (root)

---

**Require:** DAS internal structures  $parent_{v_i}$ ,  $children_{v_i}$ ,  $articulation_{v_i}$ , e  $dualLink$ ; DBS internal structures  $bridge$  and  $bridgePair_{v_i}$

**Ensure:** Each node knows, using its local array  $component_{v_i}$  the list of bi-connected component it belongs.

# Initialization of node  $v_i$

```
1:  $component_{v_i} \leftarrow \emptyset$ ;  
2: if  $v_i = root$  then  
3:   if  $articulation_{v_i} = TRUE$  then  
4:     for all  $v_j \in children_{v_i}$  do  
5:       send message  $COMPONENT(id(v_j))$  to  $v_j$ ;  
6:        $component_{v_i} \leftarrow component_{v_i} \cup \{id(v_j)\}$ ;  
7:     end for  
8:   else  
9:      $component_{v_i} \leftarrow component_{v_i} \cup \{id(v_i)\}$ ;  
10:    send message  $COMPONENT(id(v_i))$  to the only element in  $children_{root}$ ;  
11:   end if  
12: end if
```

# Node  $v_j$  receive a  $COMPONENT(c)$  message from node  $v_i$

```
13:  $component_{v_j} \leftarrow component_{v_j} \cup \{c\}$ ;  
14: if  $articulation_{v_j} = FALSE$  then  
15:   for all  $v_k \in children_{v_j}$  do  
16:     send message  $COMPONENT(c)$  to  $v_k$ ;  
17:   end for  
18: else  
19:   for all  $v_k \in children_{v_j} \mid v_k \neq dualLink_{v_j}$  do  
20:      $component_{v_j} \leftarrow component_{v_j} \cup \{id(j)\}$ ;  
21:     send message  $COMPONENT(id(v_j))$  to  $v_k$ ;  
22:   end for  
23:   if  $dualLink_{v_j} \neq 0$  then  
24:     send message  $COMPONENT(c)$  to  $dualLink_{v_j}$ ;  
25:   end if  
26: end if
```

---

de cada filho à lista de componentes a que pertence e envia uma mensagem para cada filho indicando que o identificador do próprio nó deve ser utilizado como identificador de componente biconectado (linhas 4-7). Se a raiz não é uma articulação, então possui um único filho. Neste caso, a raiz utiliza seu identificador de nó como identificador de componente biconectados e propaga este valor para o seu único filho (linhas 9-10).

Após a inicialização, o restante do algoritmo (linhas 13-26) é executado somente quando o nó recebe uma mensagem  $COMPONENT(c)$ . O valor  $c$  recebido por  $v_j$  é adicionado a lista de componentes a que o nó pertence (linha 13). Se o nó não é uma articulação, o valor  $c$  deve ser propagado a todos os seus filhos (linhas 15-17) indicando que estes nós estão no mesmo componente que  $v_j$ . Quando o nó é uma articulação, os identificadores dos nós filhos são utilizados como identificadores de componentes biconectados para determinar que, a partir daquele ponto, começa um novo componente. O identificador de cada nó filho é adicionado à lista de componentes a que a articulação pertence e

enviados ao próprio nó (linhas 19-22). Quando a articulação possui duas arestas para o mesmo componente (conforme explicado na descrição do algoritmo DAS), o identificador recebido na mensagem é propagado para o nó da aresta *dualLink* (linhas 23-25).

A Figura 4.4 exemplifica o algoritmo DDBC. A Figura 4.4(a) apresenta a entrada para o algoritmo. As linhas pontilhadas correspondem às arestas *nontree*, enquanto as setas representam a estrutura hierárquica da árvore. A articulação  $v_3$  possui duas arestas para um mesmo componente biconectado. Uma das arestas está identificada como *dualLink*. Este é o caso especial em que a articulação deve manter em sua memória, durante a execução do algoritmo DAS, uma referência para a existência de duas arestas para o mesmo componente biconectado. Na Figura 4.4(b) a raiz define o valor do seu componente com o valor de seu próprio identificador. O identificador de componente é representado com um número próximo aos nós. O identificador de componente da raiz é enviado para o nó  $v_3$  (seu único descendente) utilizando uma mensagem *COMPONENT*(1). A Figura 4.4(c) ilustra o comportamento da articulação  $v_3$ . O seu próprio identificador é enviado em uma mensagem *COMPONENT* para  $v_4$ . Este nó envia identificador recebido para  $v_2$ , já que este nó está na variável *dualLink*. A Figura 4.4(d) apresenta o comportamento para a articulação  $v_4$ . Este nó adiciona o valor recebido na lista de componentes a que participa. Em seguida, adiciona o identificador do nó filho  $v_5$  como um dos identificadores de componentes a que pertence, e então envia a este filho o seu próprio identificador como o número do componente biconectado a que pertence. A Figura 4.4(e) exemplifica o caso em que um nó comum recebe uma mensagem *COMPONENT*( $c$ ). Ele inclui o valor  $c$  na sua lista de componentes biconectados e propaga este mesmo valor para o seu filho, o nó  $v_6$ . A Figura 4.4(f) apresenta o resultado final. Neste exemplo, três componentes biconectados foram identificados. Note que os nós  $v_3$  e  $v_4$  participam de mais de um componente biconectado.

A complexidade do algoritmo DDBC pode ser computada em função do número de mensagens. Considere árvore  $T(V, E')$  previamente produzida com o algoritmo DAS a partir de um grafo  $G(V, E)$  e com as articulações devidamente identificadas. Pondere-se, também, que estão disponíveis informações sobre pontes produzidas pelo DBS. O algoritmo percorre a árvore  $T$  enviando uma mensagem por meio de cada aresta em  $E'$ . Cada nó envia uma única mensagem a cada um de seus filhos. Assim, o custo computacional do algoritmo DDBC é  $O(|V|)$  em relação à comunicação. O Lema 4.3.1 sumariza a complexidade do algoritmo.

**Lema 4.3.1.** *Considere um grafo  $G(V, E)$  e uma árvore  $T(V, E')$ ,  $E' \subset E$  criada com o algoritmo DAS. O algoritmo DDBC requer  $O(|V|)$  mensagens, no pior caso, para definir os componentes biconectados a que cada nó pertence.*

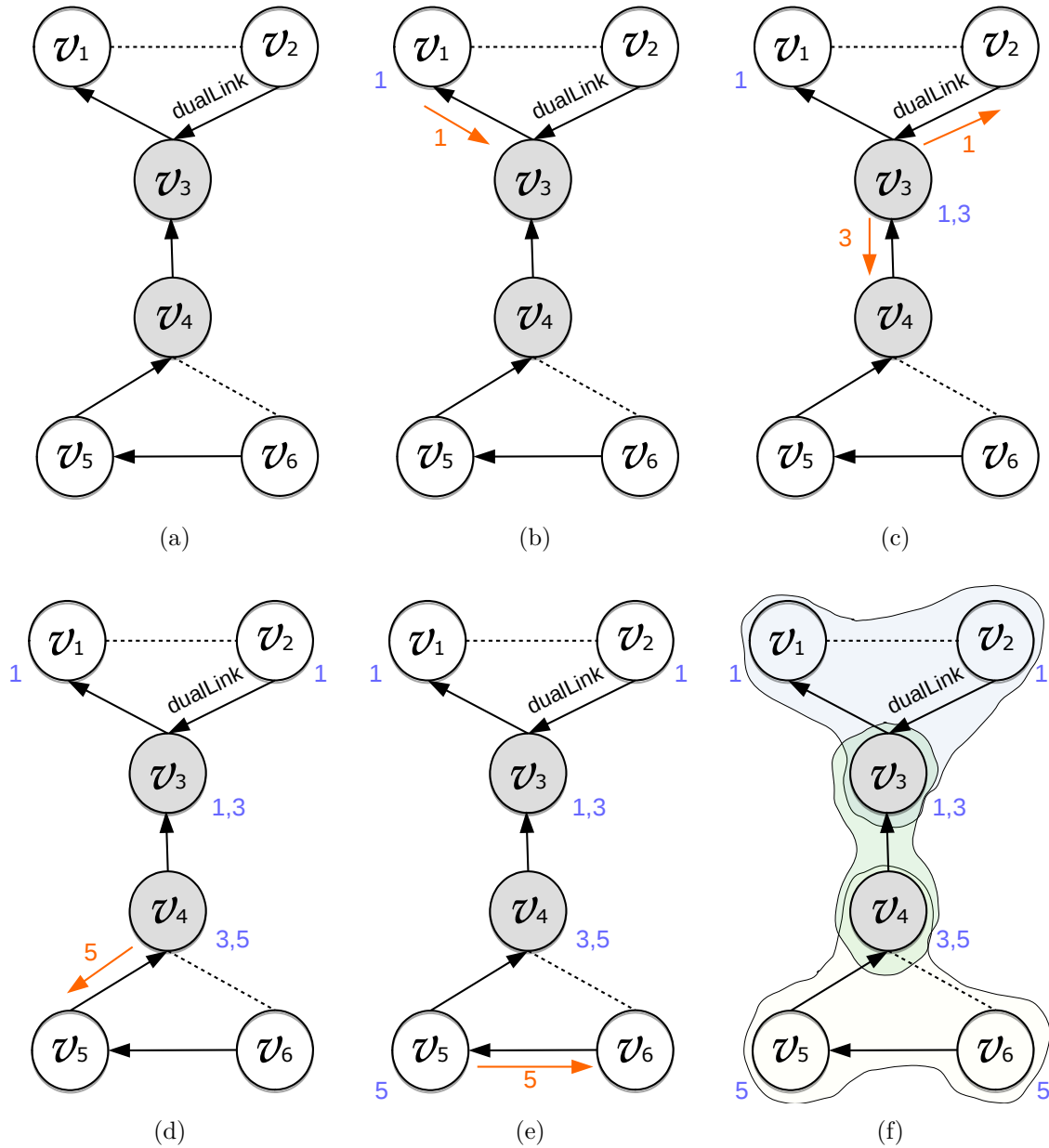


Figura 4.4: Exemplo do algoritmo DDC.

*Demonstração do Lema 4.3.1.* Considere uma árvore  $T(V, E')$  construída sobre o grafo  $G(V, E)$ , tal que  $E' \subset E$ , utilizando o algoritmo DAS. Considere também, que o algoritmo DBS foi executado nesta árvore. De acordo com os Lemas 4.1.1 e 4.2.1, cada nó sabe se é uma articulação e se possui pontes. Os algoritmos ainda permitem que cada nó saiba, por meio de variáveis locais, a identificação do nó pai e dos nós filhos na árvore e os vizinhos incidentes a cada ponte. Assuma que um nó  $v_j$  enviou uma ou mais mensagens *COMPONENT* para o nó pai  $v_i$  por meio do enlace  $v_i v_j$ . Temos uma contradição, pois o algoritmo garante que mensagens são enviadas apenas para nós filhos (linhas

4, 10, 15, 19, 23). Assuma que  $v_j$  recebeu duas mensagens de  $v_i$ . É uma contradição, pois o nó  $v_i$  encerra a execução do algoritmo após enviar as mensagens para os nós filhos. O algoritmo DDBC envia, para cada enlace em  $E'$ , uma única mensagem. Como a árvore  $T$  possui  $|V| - 1$  arestas, o custo computacional em termos de comunicação é  $O(|V|)$ .  $\square$

## 4.4 Resultados Empíricos

As propostas apresentadas neste capítulo foram implementados no Matlab<sup>®</sup>. Os parâmetros de simulação são similares aos utilizados por Yu *et al.* [61] e Zhu *et al.* [62]. Os nós possuem uma potência máxima de transmissão de  $P_{MAX} = 4900$  unidades de potência, o fator de desvanecimento de sinal é definido como  $\alpha = 2$  e o limiar de potência recebida de  $\tau = 1$ . Isso equivale a um raio de transmissão de  $R_{MAX} = 70m$ . A densidade é variada de  $4 \times 10^{-4}$ ,  $4,5 \times 10^{-4}$ , ...,  $12 \times 10^{-4}$  em uma área bidimensional de  $300m \times 300m$ . A potência de transmissão foi variada de tal forma a obter raios de transmissão de  $20m$ ,  $30m$ , ...,  $70m$  de alcance individual para todos os nós. O nível de significância para os dados apresentados é de 95%.

Os algoritmos DAS, DBS e DDBC foram submetidos a um conjunto de 1.000 topologias geradas de forma aleatória. Foram verificados, após a execução dos algoritmos, a quantidade de articulações, pontes e o número de articulações que não formam pontes. Para validar os resultados produzidos, o algoritmo DDFS e aqueles propostos Chaudhuri [13, 14] também foram aplicados aos gráficos. Os dados foram comparados, durante as etapas de desenvolvimento, para depurar as soluções propostas. Os resultados corroboram que os algoritmos propostos estão corretos bem como suas implementações.

O principal objetivo deste trabalho é apresentar mecanismos para recuperação de conectividade por meio da Comunicação Cooperativa. Os elementos críticos são utilizados para prever a segmentação da rede, enquanto a identificação dos componentes biconectados fornecem mecanismos para identificar a topologia após a segmentação. Assim, os resultados apresentados nesta seção têm como alvo a definição dos cenários em que melhor se aplicam as soluções para recuperação da conectividade. Os dados são utilizados para definir a potência de transmissão e a densidade para os algoritmos de recuperação de conectividade propostos no Capítulo 5, bem como para explicar o desempenho destes algoritmos em função do número de articulações e pontes em uma rede *ad hoc* sem fio.

A Figura 4.5 ilustra os resultados obtidos ao aplicar os algoritmos *DAS* e *DBS*. A topologia é um grafo UDG gerado de forma aleatória em uma área de  $300 \times 300m$ , com densidade  $5,5 \cdot 10^{-4}$ . Todos os nós utilizam a potência de transmissão máxima, isto é,  $P_i = P_{MAX} = 4900$ . Os nós são representados por pontos e os enlaces por linhas. A Figura 4.5(a) apresenta a topologia da rede, composta por dois componentes biconectados

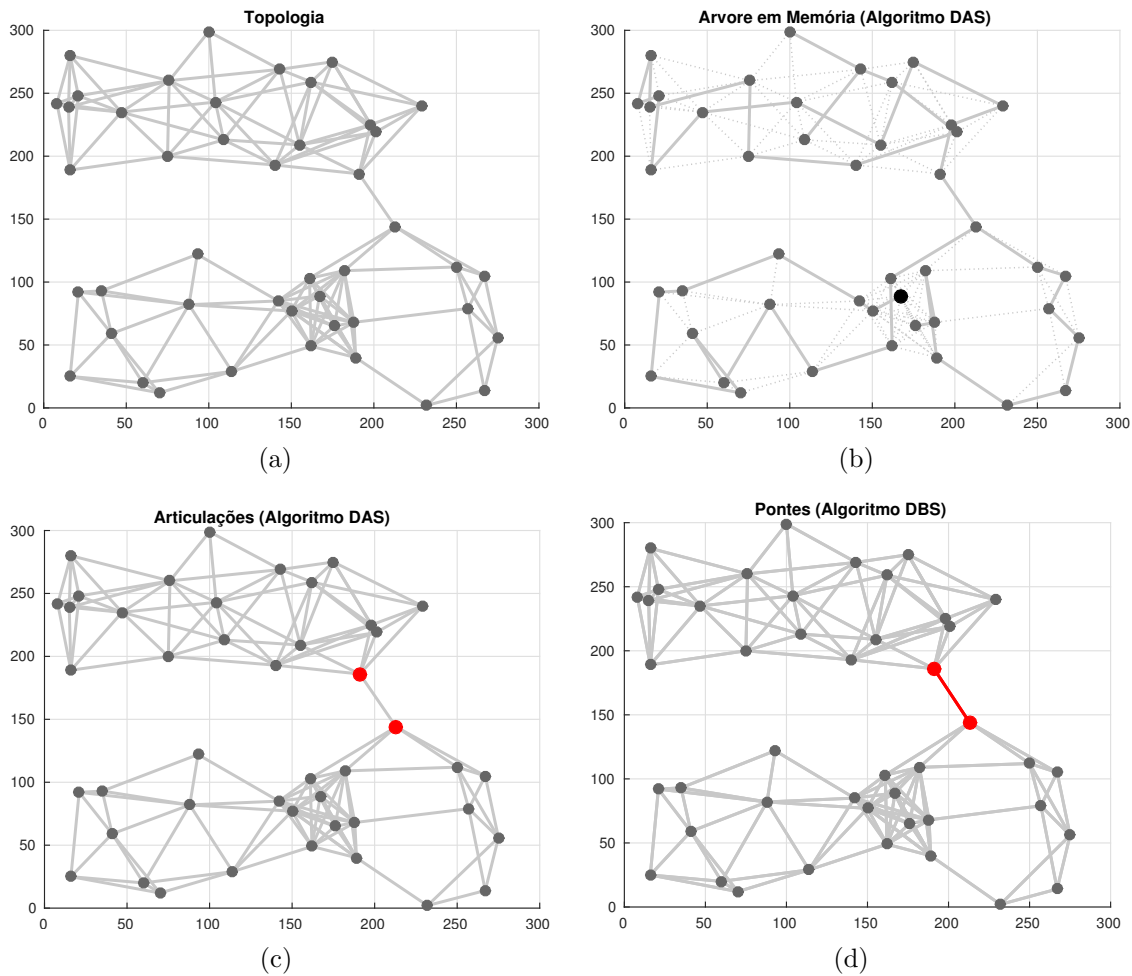


Figura 4.5: Exemplo da localização dos elementos críticos e dos componentes. (a) grafo UDG obtido com  $P_{MAX}$ ; (b) árvore distribuída; (c) articulações; e (d) articulações e pontes.

ligados por uma ponte. A Figura 4.5(b) apresenta a árvore criada em memória de forma distribuída, utilizando o algoritmo *DAS*. O nó raiz foi escolhido aleatoriamente e está destacado por meio de um ponto com maior dimensão. As linhas sólidas apresentam as arestas da árvore, enquanto as pontilhadas demonstram as arestas *non-tree* (arestas removidas do grafo para obter a árvore). A Figura 4.5(c) ilustra o grafo após o fim da execução do algoritmo *DAS*. As duas articulações encontradas estão ilustradas por meio de pontos vermelhos. A Figura 4.5(d) ilustra o grafo após a execução do algoritmo *DBS*. A ponte encontrada está destacada por uma linha vermelha.

A Figura 4.6 apresenta a porcentagem de grafos conectados. O eixo “z” representa a porcentagem de grafos conectados, o eixo “x” corresponde à densidade do grafo e o eixo “y” ao raio de transmissão. Podemos notar que a porcentagem de grafos conectados aumenta

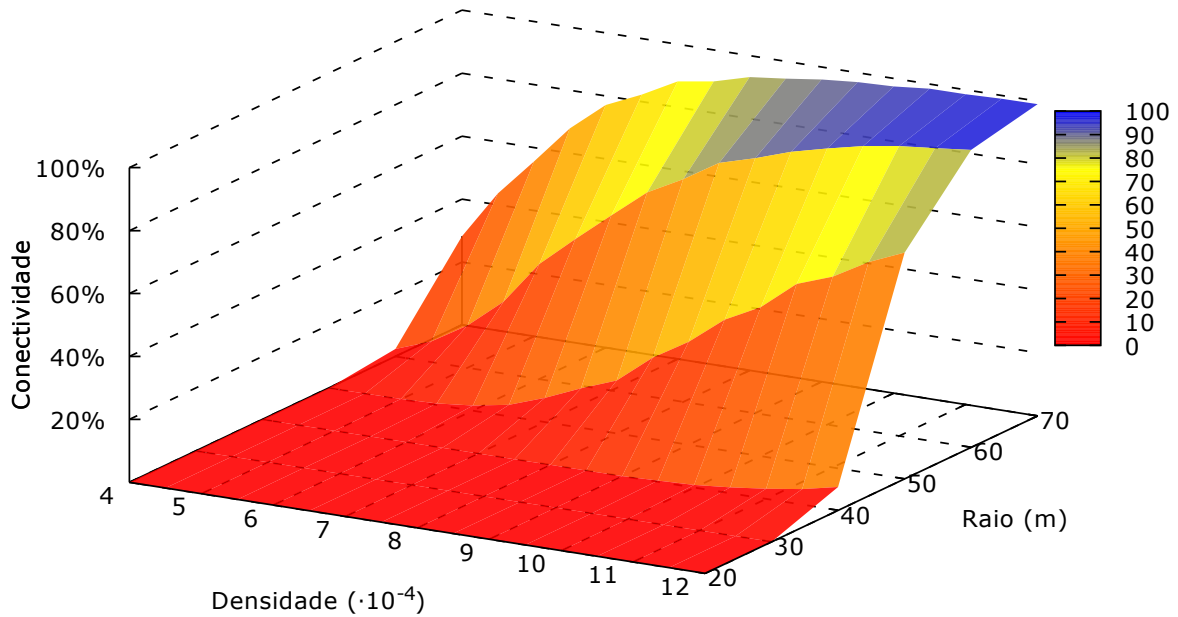


Figura 4.6: Quantidade de grafos conectados em função do raio de transmissão e densidade.

rapidamente com incremento na densidade e no raio de transmissão, quando for igual ou inferior a  $30m$ , não é possível obter grafos conectados. Com raio de transmissão entre  $40m$  e  $60m$ , o número de grafos conectados cresce rapidamente com o aumento na densidade. Quando o raio de transmissão é de  $70m$  (com a potência máxima de transmissão) a quantidade de grafos conectados é superior a  $90\%$ , em média. A porcentagem de grafos conectados é, em média,  $5,9\%$  para a densidade de  $4 \times 10^{-4}$ . Com densidade inferior a  $7,0 \times 10^{-4}$ , a porcentagem de grafo conectados é de  $13,57\%$ , em média. O valor é superior a  $99\%$  quando a densidade é  $12 \times 10^{-4}$ . Para o maior valor de densidade simulado, espera-se um grande número de articulações e pontes já que o grafo ainda é esparso com um conjunto de nós que abrangem, com o raio de comunicação máximo, apenas  $30\%$  da área.

A Figura 4.7 determina a quantidade de articulações e pontes para os grafos conectados. O raio de transmissão variou entre  $55, 60, 65$  e  $70m$ . O eixo “y” corresponde à quantidade média de elementos críticos encontrados, enquanto o eixo “x” apresenta a densidade e o eixo “y” o raio de transmissão. Nesta figura, são apontadas as médias para os números de articulações (linhas pretas), para o número de pontes (linhas azuis) e quantidade de articulações que não formam pontes (linhas vermelhas). Ao observar a figura, podemos notar que a quantidade de elementos críticos reduz rapidamente com o aumento na densidade ou no raio de transmissão. Em todos os casos, o número de articulações que não participam de pontes é superior ao número de pontes. O número médio de articulações cai de  $27,61$  unidades para  $0,8$  quando o raio de transmissão é de



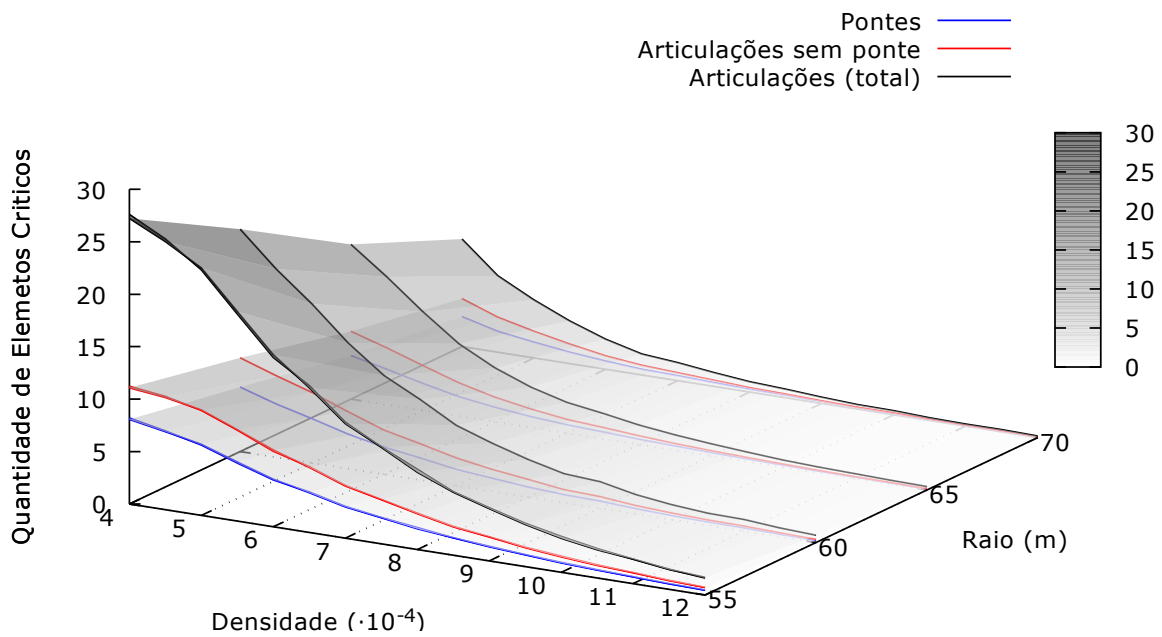


Figura 4.7: Quantidade de articulações e pontes por densidade.

55m. Este número reduz de 10,27 para 0,14 quando o raio de transmissão é de 70m. Este comportamento é recorrente em todos os cenários simulados, indicando que, quanto maior o raio, menor a probabilidade de ocorrerem articulações no grafo. Quando a densidade é de  $4 \times 10^{-4}$ , o número médio de articulações é de 27,61 para o raio de 55m e decai para 10 com raio de 70. Como esperado, este comportamento indica que o número de articulações decai também em função do aumento na densidade. No pior caso, o número de pontes é de 8,19, enquanto o melhor cenário apresenta 0,41 ponte. Para os mesmos cenários, o número de articulações que não formam pontes reduz, respectivamente, de 11,22 para 0,59. A quantidade de pontes decaiu 95%, à medida que o número de articulações que não forma pontes decaiu 94,75%. Os dados apresentados indicam que os cenários avaliados são suscetíveis à presença de um grande número de elementos críticos.

A Figura 4.8 exibe a quantidade de elementos críticos quando o raio de transmissão é de 70m. O eixo “x” corresponde à densidade do grafo, ao passo que o eixo “y” apresenta a quantidade de articulações e pontes. Nesta figura, a área cinza representa a quantidade total de articulações, a linha marcada com pontos à quantidade de articulações que não formam pontes e a linha marcada com quadrado à quantidade de pontes. A porcentagem de nós críticos na rede, em relação ao total de nós, reduziu de 28,54%, no pior cenário, para 0,13% no melhor cenário. A porcentagem de articulações que não formam pontes, em função do número total de elementos críticos, é de 61,41% com intervalo de confiança de  $\pm 0,012$ . Isto indica que, para este cenário, a relação entre o número de componentes ligados por articulações e pontes não depende da densidade ou raio de transmissão.

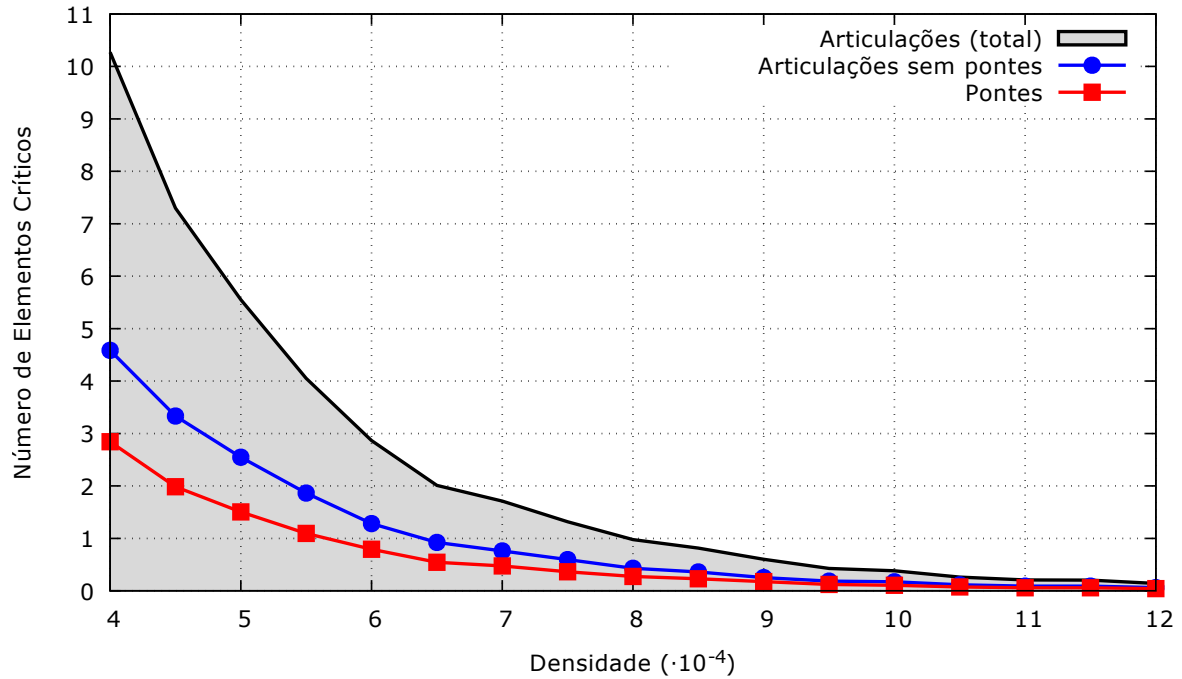


Figura 4.8: Quantidade de articulações e pontes por densidade.

Em [9, 31, 37, 43], os autores demonstram que a conectividade da rede e a densidade dos nós está diretamente relacionado ao raio de transmissão e sua área de cobertura. Resultados analíticos demonstram que a densidade dos nós não é homogênea em toda a rede [31, 43]. Em especial, o efeito de borda introduz uma não linearidade no grau dos nós. Os nós próximos à borda da rede tendem a possuir menor grau de conectividade. Nunes e Laranjeira [31] apresentam mecanismos para minimizar o efeito de borda em redes *ad hoc* sem fio. Considerando estes aspectos, foi possível notar diversos casos em que um único nó é ligado à rede por meio de uma ponte. Estes cenários são desfavoráveis à recuperação de conectividade, isto é, consideramos neste trabalho que não é viável recuperar a conectividade para um único nó da rede. Tornando-se ainda mais relevante considerando a técnica de Comunicação Cooperativa utilizada, que requer a cooperação de vizinhos para restabelecer a conectividade da rede.

Na Figura 4.9, são apresentados os números relativos de pontes e de articulações quando o efeito de borda é considerado. Nesta figura, as barras sólidas indicam o número relativo de articulações que não formam pontes, à medida em que a barra hachurada corresponde à porcentagem de pontes. A porcentagem de componentes ligados por pontes reduz de, aproximadamente, 56% para 11%, ao passo que a porcentagem de componentes ligados por articulações aumenta de 44% para 89%. Ao comparar com a Figura 4.8, podemos notar que a relação entre articulações que não formam pontes e o número de pontes é diferente. Os dados indicam que, quanto maior a densidade da rede, maior é

a probabilidade dos componentes se conectarem por meio de articulações. Este fato é corroborado por dados analíticos que demonstram que, com aumento na densidade (ou redução na área em que os nós são dispostos), menor é a distância média entre os nós [37]. Com nós mais próximos, maior é a probabilidade de se encontrar componentes biconectados ligados por articulações do que por pontes. De fato, a distância máxima entre os componentes de rede é de  $2 \times R_{MAX}$  quando ligados por uma articulação e  $3 \times R_{MAX}$  quando ligados por uma ponte.

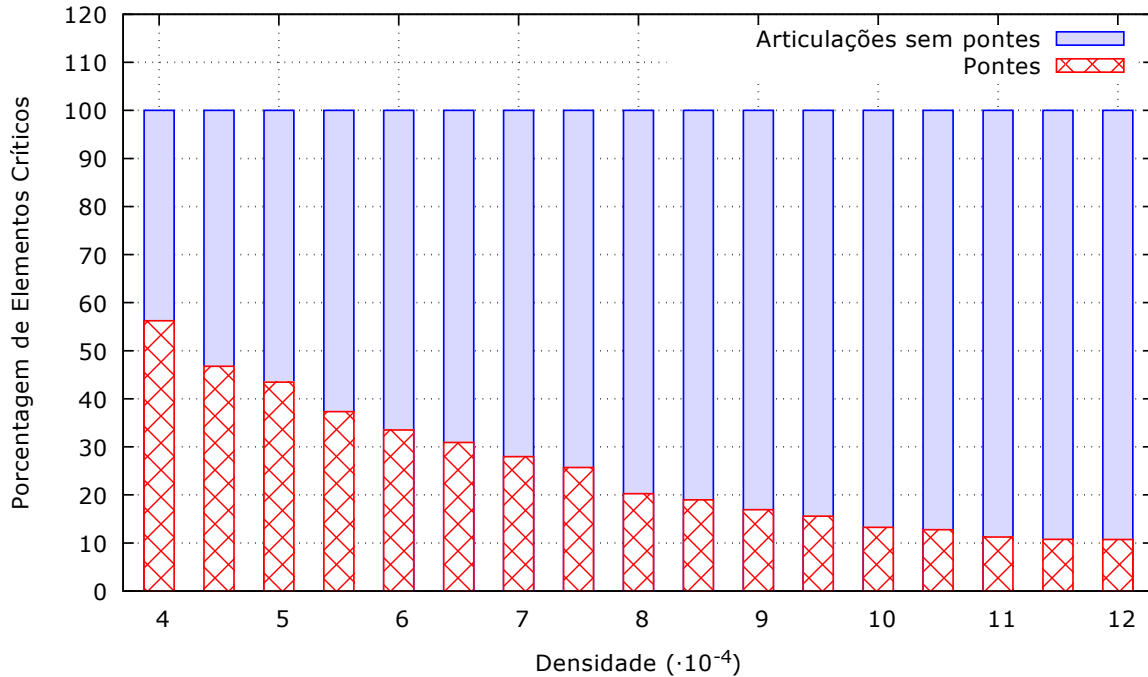


Figura 4.9: Quantidade de articulações e pontes para  $d = 70$  em uma área de  $300 \times 300m$ .

## 4.5 Síntese e Discussão

Este capítulo apresentou um conjunto de algoritmos para localizar os elementos críticos e os componentes biconectados de uma rede *ad hoc* sem fio. O algoritmo *Distributed Articulation Search* (DAS) constrói, a partir de um grafo  $G(V, E')$ , uma árvore distribuída  $T(V, E')$ ,  $E' \subset E$ . A estrutura da árvore é utilizada para determinar as articulações do grafo por meio das premissas elaboradas por Hopcroft e Tarjan [22]. O algoritmo *Distributed Bridge Search* (DBS) utiliza o conhecimento produzido pelo DAS para determinar as pontes do grafo. Por fim, o algoritmo *Distributed Define Biconnected Components* (DDBC) atribui, com base nas informações sobre articulações e pontes, um identificador para cada componente biconectado do grafo. A informação é armazenada de forma distribuída, permitindo que cada nó saiba qual o componente a que pertence.

O *DAS* utiliza como base o algoritmo DDFS, assim como a proposta de Chaudhuri [14]. O algoritmo de Chaudhuri [14] tem um custo computacional de  $O(|V|)$  mensagens para localizar as articulações da rede e utiliza um total de  $4 \times |V|$  mensagens e quatro tipos diferentes de mensagens. Apesar do custo computacional de ambas as propostas serem iguais, o *DAS* reduz pela metade o número de mensagens necessárias e utiliza somente um tipo de mensagem. Outra contribuição importante está na redução da quantidade de dados trafegados nas mensagens. O *DAS* utiliza conceitos básicos da teoria de conjuntos para evitar a repetição de identificadores de nós em cada mensagem.

Os algoritmos *DBS* e *DDBC* são capazes de determinar as pontes e os componentes utilizados numa pequena quantidade de mensagens de controle. No pior cenário, ambos requerem  $O(|V|)$  mensagens. A metodologia utilizada por esses algoritmos é diferente das soluções anteriores [13]. A estratégia utilizada permite realizar estas tarefas com um esforço menor, já que utilizam informações sobre a árvore distribuída e sobre articulações para restringir o processo de busca por pontes e definição dos componentes biconectados.

Os dados apresentados neste capítulo têm como objetivo definir os cenários em que melhor se aplicam as soluções para recuperação de conectividade. Os parâmetros de simulação foram limitados àqueles utilizados por trabalhos relacionados à conectividade de rede [61, 62]. Os dados demonstram que o número de articulações e pontes dos grafos conectados são afetados por variações na densidade e no raio de transmissão.

Observando trabalhos que tem como objetivo garantir a conectividade inicial da rede, por meio do aumento no alcance máximo dos nós [36, 60, 61, 62], podemos notar que os parâmetros de simulação são referentes a regiões em que a porcentagem de grafos conectados é pequena. Considerando estes dados empíricos, pode-se notar que a recuperação de conectividade torna-se propícia quando o raio de transmissão é superior a  $60m$ . Neste caso, espera-se que a topologia é conectada e tende a apresentar um grande número de elementos críticos. Além disso, durante a operação de uma rede sem fio, a densidade decresce, por exemplo, em função do esgotamento da bateria dos nós. Assim, mesmo que a rede seja implantada com parâmetros que garantam a conectividade da rede e baixa probabilidade de existirem elementos crítico, os nós tendem a falhar, resultando na presença de articulações.

Conforme os dados empíricos apresentados neste capítulo, percebemos que o efeito de borda tem impacto direto na quantidade e na relação entre o número de articulações e pontes. Este resultado pode evitar que se dispense um grande esforço para recuperar a conectividade de nós individuais ligados à rede por uma ponte. Espera-se que mecanismos para a recuperação de conectividade devam desprender mais esforço para recuperar a conectividade quando uma ponte falha em relação ao caso em que uma articulação falha em função da distância média entre os componentes. Assim, com o aumento na densidade,

menor é o número relativo de pontes e, portanto, menor será o custo médio para recuperar a conectividade.

Como discutido no Capítulo 2, a localização dos elementos críticos permite prever a segmentação da rede e, assim, disponibilizar mais nós ou mover os existentes para reforçar a conectividade. Porém, é possível utilizar a Comunicação Cooperativa para garantir a conectividade da rede já que esta técnica permite a criação de enlaces além do alcance máximo dos nós. Em seguida, o Capítulo 5 apresenta duas soluções distribuídas para recuperar a conectividade, aplicando a Comunicação Cooperativa, baseado nas informações mantidas pelo DAS, DBS e DDBC.

# Capítulo 5

## Recuperação de Conectividade

Este capítulo apresenta dois mecanismos distribuídos para permitir que os nós de articulação e os seus vizinhos possam coordenar e encontrar alternativas para restabelecer a conectividade de uma rede *ad hoc* sem fio. Em linhas gerais, estas soluções consistem em utilizar informações localizadas a um ou dois saltos para encontrar, de forma preventiva, enlaces cooperativos entre os componentes biconectados ligados por articulações.

A primeira solução permite que as articulações calculem os enlaces cooperativos sobressalentes entre nós de diferentes componentes e os notifiquem para recuperar a conectividade quando necessário. Esta solução requer informações sobre a topologia a 2 saltos e uma estimativa de distância entre os nós. Em caso de falha do nó crítico, a conectividade é restabelecida utilizando os enlaces cooperativos previamente identificados. Na segunda solução, os vizinhos das articulações calculam os enlaces cooperativos empregando estimativas de distância arbitrária entre os componentes. Neste caso, não é necessário coletar informações adicionais sobre a topologia da rede. A distância entre os componentes é sobre-estimada permitindo a criação de múltiplos enlaces que são eliminados, caso necessário, a partir de um ajuste na potência de transmissão. Para selecionar os nós auxiliares, são propostas duas heurísticas simples, que tem como principal objetivo reduzir o tempo de execução, ao contrário de soluções similares, que tem como finalidade, por exemplo, a distribuição da potência entre o maior número de nós.

O restante deste capítulo é organizado como segue. A Seção 5.1 apresenta a solução distribuída, utilizando informações de dois saltos para criar enlaces redundantes. A Seção 5.2 expõe uma evolução do algoritmo que reduz o custo computacional e não requer a coleta de dados sobre a topologia da rede, isto é, usa apenas informação a um salto. E, finalmente, a Seção 5.3 resume os resultados das técnicas apresentadas para a recuperação de conectividade.

## 5.1 Solução Distribuída e Localizada a 2 Saltos

Esta seção apresenta uma solução distribuída e localizada a 2 saltos para a recuperação da conectividade da rede por meio de enlaces cooperativos. Todos os nós executam o mesmo algoritmo para a recuperação de conectividade. Contudo, existem dois gatilhos para que ações sejam realizadas: um nó é definido como articulação e/ou uma mensagem é recebida indicando que o nó deve criar um enlace cooperativo. A identificação dos nós críticos deve ser, preferencialmente, determinada por algoritmos distribuídos, considerando que cada nó deve saber, por meio de variáveis locais, se são articulações. As informações localizadas são necessárias apenas nas articulações e podem ser coletadas utilizando mensagens especiais ou trocas de informações sobre a tabela de roteamento.

A recuperação da conectividade consiste em calcular, nas articulações, o melhor enlace cooperativo entre seus vizinhos em cada um dos componentes ligados por ela. Assim, informações a dois saltos são necessárias para determinar o conjunto de auxiliares. Para a seleção dos auxiliares, uma heurística simples é proposta. Em seguida, os nós escolhidos são notificados por meio de uma mensagem. Ao receber a mensagem, os vizinhos escolhidos monitoram o elemento crítico e, quando este se torna indisponível, o enlace é efetivamente criado. A Figura 5.1 ilustra a recuperação de conectividade por falha em articulações e pontes. As arestas e nós pontilhados representam os elementos removidos da topologia. Na Figura 5.1(a), uma articulação identificada por  $v_a$ , que liga dois componentes biconectados, falha. O enlace cooperativo  $\widetilde{v_i v_j}$  é mantido por dois nós, um em cada componente, para recuperar a conectividade. Na Figura 5.1(b), a ponte  $v_a v_b$  deixa de existir por falha na articulação  $v_a$ . A recuperação da conectividade é realizada por meio do enlace cooperativo  $\widetilde{v_i v_b}$  entre o nó  $v_i$  e a articulação remanescente  $v_b$ . A seção a seguir apresenta, em detalhes, a solução.

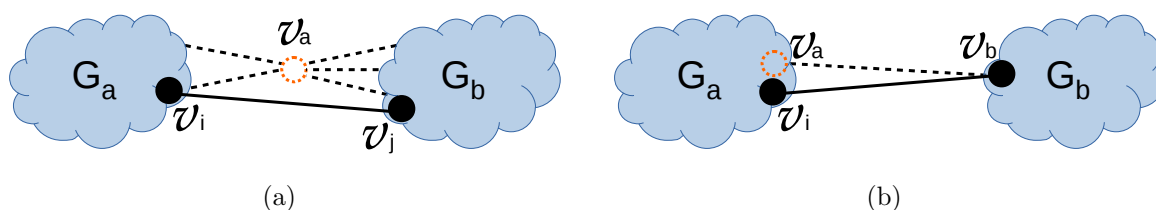


Figura 5.1: Utilização do CARCC2h para recuperar a conectividade para falha em: (a) articulação; e (b) ponte.

### 5.1.1 Detalhamento da Solução

A solução proposta tem como premissa a recuperação da conectividade em tempo hábil, durante a operação da rede. Para reduzir o custo computacional, utiliza-se uma heurística

---

**Algorithm 5** DBHSS( $v_i, v_j$ )

---

**Require:** Source node  $v_i$ , destination node  $v_j$ .

**Ensure:** The smaller helper set  $H_{i,j}$  complying with Eq. (3.3) and the required transmission power

$W_{CC}(H_{i,j})$  to create a cooperative link from  $v_i$  to  $v_j$ ;

1:  $\forall k \in N(v_i), W_i = d_{v_i,k} + d_{v_j,k}$ ;

2:  $C \leftarrow N(v_i)$  sorted by  $W_i$ ;

3:  $H_{i,j} \leftarrow \emptyset$

4: **repeat**

5:   Remove first node from  $C$  and add to  $H_{i,j}$

6:   Node  $v_i$  compute  $P^r$  of  $v_i v_j$  and  $W_{CC}(H_{i,j})$  using Equation 3.3;

7: **until**  $P^r < \tau$  and  $C \neq \emptyset$ ;

8: **if**  $P^r \geq \tau$  **then**

9:   return  $H_{i,j}$  and required transmission power;

10: **else**

11:   return  $H_{s,d} = \emptyset$  and  $W_{CC}(H_{i,j}) = 0$ ;

12: **end if**

---

simples para selecionar os nós auxiliares. A rotina, chamada *Distance Based Helper Set Selection* (DBHSS) ou Seleção de Auxiliares Baseada na Distância, é apresentada em detalhes no Algoritmo 5. A ideia geral é adicionar a menor quantidade possível de nós ao conjunto de auxiliares, utilizando uma métrica simples, até que o enlace cooperativo possa ser estabelecido. O objetivo é obter o valor mínimo a soma da distância entre os auxiliares e os nós de origem e destino, isto é:

$$\min(\sum(d_{k,i} + d_{k,j})), k \in H_{i,j}, \quad (5.1)$$

tal que a conectividade pode ser estabelecida com o conjunto  $H_{i,j}$  de tamanho mínimo. Dado uma origem  $v_i$  e um destino  $v_j$ , inicialmente, os candidatos a auxiliares  $k \in N(v_i)$  são ordenados de forma crescente em função da soma da distância até a origem  $d_{v_i,k}$  e até o destino  $d_{v_j,k}$  (linhas 1-2). Em seguida, os nós são adicionados, gradualmente, ao conjunto de auxiliares  $H_{i,j}$  até que um enlace cooperativo possa ser estabelecido (linhas 3-7). O enlace pode ser estabelecido quando a potência recebida no destino  $P^r$  é maior ou igual ao limiar de detecção do destino  $\tau$ . A heurística retorna o conjunto de auxiliares e o custo do enlace cooperativo, quando for possível estabelecer o enlace (linhas 8-9). Se todos os vizinhos são adicionados e não é possível estabelecer o enlace cooperativo, a função retorna um conjunto vazio (linhas 10-12).

A conectividade é recuperada pelo Algoritmo *Connectivity Auto-Recovering through Cooperative Communication Using 2-hop Information* (CARCC2h) ou Recuperação Automática de Conectividade Por Meio da Comunicação Cooperativa Utilizando Informações a 2 Saltos. Como requisito, a solução considera que cada nó sabe se é uma articulação, que é possível coletar informações a dois saltos a partir da articulação e que podem ser calculadas estimativas de distância entre pares de nós. Os nós podem determinar se



são articulações ao executar os algoritmos propostos por Chaudhuri [14] e Jorgic [24] ou o algoritmo DAS apresentado no Capítulo 4. As informações sobre a topologia podem ser obtidas utilizando mensagens especiais. Protocolos de roteamento, como o *Ad Hoc On-Demand Distance Vector* (AODV) [41], utilizam uma mensagem especial, chamada “*hello*”, para coletar informações de um salto [42]. De forma similar, as informações de dois saltos podem ser obtidas pela troca de informações diretamente entre os nós ou por meio da troca de tabela de roteamento. A articulação pode determinar se os nós estão em um mesmo componente ao verificar a existência de rotas entre eles. Os algoritmos DBS e DDBC podem ser aplicados para determinar os componentes biconectados a que cada nó pertence. Esta informação é encaminhada para a articulação ou coletada juntamente com as informações topológicas. A potência de transmissão pode ser, explicitamente, informada nas mensagens “*hello*” ou por meio da troca das informações de roteamento. Com estas informações e as características básicas do modelo de comunicação considerados neste trabalho, as estimativas de distância entre os nós pode ser obtida. De fato, foram relatadas na literatura a estimativa de distância baseado no RSSI ou *Received Signal Strength Indicator*, com precisão razoável [57]. O trabalho de Mikko Kohvakka *et al.* [27] demonstrou que o desvanecimento de sinal pode ser determinado por quadros transmitidos em diferentes potências. Com informações sobre o desvanecimento de sinal e a potência de transmissão, pode-se estimar a distância mesmo se o RSSI não é conhecido [27]. Além disso, o erro de precisão nas estimativas de distância podem ser compensados por meio de um pequeno incremento na potência de transmissão. O objetivo é obter o enlace cooperativo de menor custo, isto é:

$$\min(w_d(H_{i,j}) + (|H_{i,j}| + 1)w_{CC}(H_{i,j})), \quad (5.2)$$

utilizando apenas informações topológicas a dois saltos. Como saída, o algoritmo determina os nós responsáveis por criar os enlaces cooperativos, seus auxiliares e a potência necessária. O CARCC2h é apresentado em detalhes, no Algoritmo 6.

O algoritmo é executado em cada nó da rede e, tem como gatilho a definição do nó como articulação ou a chegada de uma mensagem do próprio algoritmo. Quando um nó é definido como articulação, as linhas 2-15 do algoritmo são executadas. A articulação calcula o enlace cooperativo de menor custo, por meio do DBHSS entre cada par de nó vizinho nos componentes interligados por ela (linhas 5-11). Em seguida, uma mensagem é enviada para os dois nós escolhidos, indicando que, caso a articulação falhe, o enlace cooperativo deve ser criado em conjunto com o outro nó (linha 12-13). Note que, se a articulação faz parte de uma ponte, então o enlace de menor custo é criado entre a articulação adjacente e um vizinho em seu próprio componente. Os enlaces cooperativos são, efetivamente, criados nas linhas 16-24. Os nós devem aguardar a mensagem *RECOVER(id)*

---

**Algorithm 6** CARCC2h()

---

**Require:** Boolean variable  $articulation_{v_r}$ , from DAS algorithm;

**Ensure:** A cooperative link  $v_i v_j$  such that  $component(v_i) \neq component(v_j)$ ;

```
1: Let  $v_r$  be the node running the algorithm;
# Articulation notify neighbors to create the backup link.
2: if ( $articulation_{v_r} = TRUE$ ) then
3:   Let  $V_i$  and  $V_j$  be the set of neighbors of the articulation  $v_r$  in each network component;
4:   for each node pair of nodes  $v_i, v_j$ ,  $v_i \in V_i$  and  $v_j \in V_j$  do
5:     Run DBHSS to estimate the cost of link  $\widetilde{v_i v_j} + \widetilde{v_j v_i}$ ;
6:   end for
7:   Let  $v_k$  and  $v_l$  be the nodes with lowest cost  $\widetilde{v_i v_j} + \widetilde{v_j v_i}$ ;
8:   Send message RECOVER( $v_k$ ) to node  $v_l$ ;
9:   Send message RECOVER( $v_l$ ) to node  $v_k$ ;
10: end if
# Nodes monitor the articulation state and create the link when it is unavailable
11: Wait for RECOVER( $id$ ) message;
12:  $v_a \leftarrow$  the source of the message (articulation);
13:  $v_b \leftarrow$  the node with identification  $id$ ;
14: Monitor if  $v_a$  is available;
15: if  $v_a$  is unavailable then
16:   Create the CC link  $\widetilde{v_r v_b}$ ;
17: end if
```

---

(linha 16) que indica que o nó recebedor da mensagem deve criar um enlace cooperativo entre ele e o nó  $v_b$ , quando a origem da mensagem estiver indisponível. O nó monitora o estado da articulação (linha 20) e cria o enlace quando necessário (linhas 20-22).

A Figura 5.2 exemplifica a recuperação de conectividade por falha em articulação. A Figura 5.2(a) mostra a topologia inicial. Na Figura 5.2(b), o nó de articulação  $v_4$  notifica aos nós  $v_2$  e  $v_6$  que devem criar um enlace cooperativo em caso de falha. Na Figura 5.2(c), a articulação falha e a rede se torna desconectada. Na Figura 5.2(d), o enlace cooperativo entre os componentes é criado. Neste caso, o enlace  $\widetilde{v_2 v_6}$  criado é uma ponte. Assim, os dois nós que integram este enlace são novas articulações. A Figura 5.3 apresenta a sequência de eventos para a recuperação de conectividade para falhas em pontes. A Figura 5.3(a) apresenta o grafo inicial. Os nós  $v_4$  e  $v_5$  são articulações que compõem a ponte. Na Figura 5.3(b) a articulação  $v_4$  calcula o melhor enlace para recuperar a conectividade em caso de falhas e informa os nós  $v_2$  e  $v_3$  escolhidos. De forma similar, a articulação  $v_5$  notifica aos nós  $v_4$  e  $v_7$  que devem recuperar a conectividade em caso de falha. A Figura 5.3(c) ilustra quando a articulação  $v_4$  falha. Na Figura 5.3(d), um enlace cooperativo é estabelecido entre os nós  $v_2$  e a articulação  $v_5$  remanescente. O enlace criado é uma ponte e, portanto, uma nova articulação aparece no grafo. A conectividade do grafo pode ser recuperada continuamente pelo mesmo algoritmo.

Para calcular o custo computacional do CARCC2h, primeiro considere a heurística de seleção de auxiliares. Inicialmente, a rotina DBHSS deve calcular uma métrica para cada vizinho da origem. Esta tarefa pode ser realizada em  $O(\Delta(G))$  tempo, onde  $\Delta(G)$

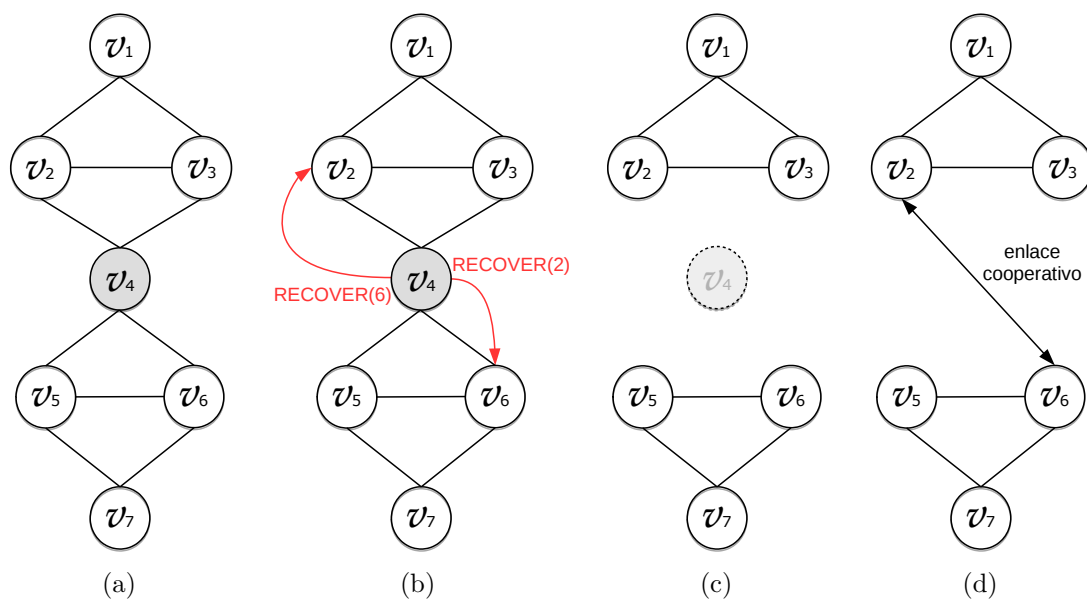


Figura 5.2: CARCC2h: Recuperação de conectividade por falha em articulação.

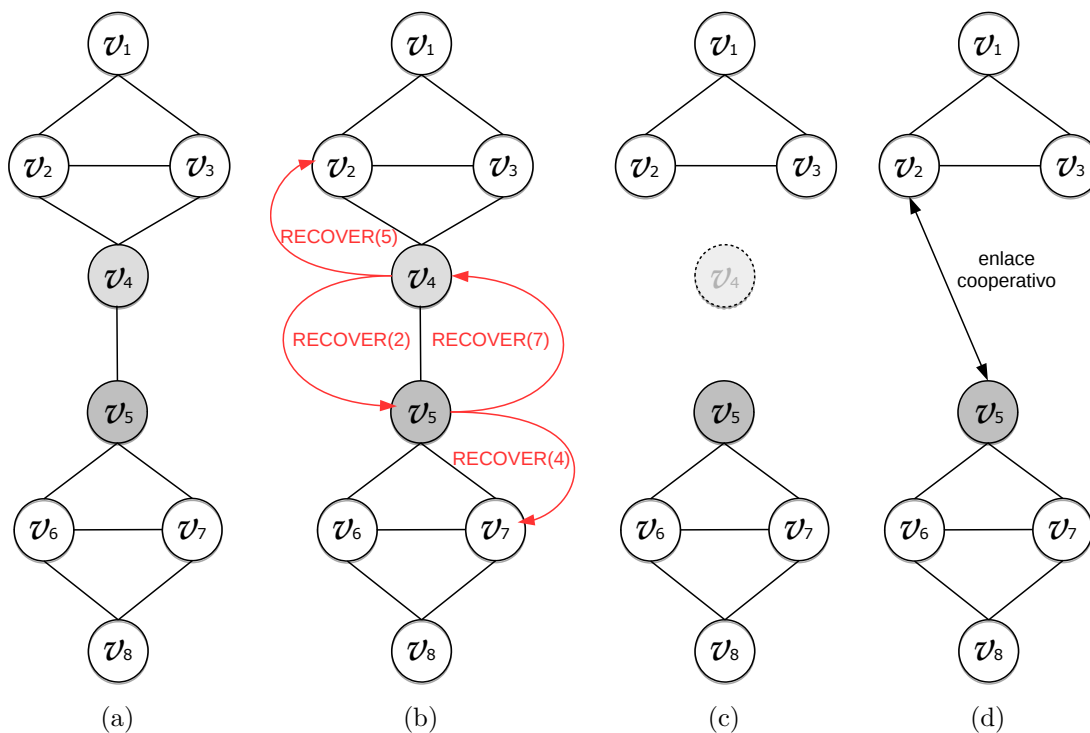


Figura 5.3: CARCC2h: Recuperação de conectividade por falha em ponte.

é o grau do grafo. A ordenação dos candidatos a auxiliares pode ser computada em  $\Delta(G) \times \log \Delta(G)$  tempo, considerando métodos eficientes de ordenação. Em seguida, os nós são adicionados ao conjunto de auxiliares até que o destino seja alcançado. Esta tarefa é realizada em  $O(\Delta(G))$  tempo. Assim o custo total da heurística DBHSS é

$O(\Delta(G) + \Delta(G) \log \Delta(G))$ . Em relação ao algoritmo de recuperação de conectividade CARCC2h, observe que a rotina de seleção de nós é utilizada na linha 5. Inicialmente, os vizinhos devem ser classificados quanto ao componente a que pertencem requerendo  $O(\Delta(G))$  tempo. A rotina de seleção de nós é empregada duas vezes para cada par de nós, totalizando  $2 \times \Delta(G)^2$ . As demais atividades do algoritmo podem ser realizadas em tempo  $O(\Delta(G))$ . Assim, o custo do CARCC2h é  $O(\Delta(G)^2 \cdot \Lambda)$ , onde  $\Lambda$  corresponde ao custo para selecionar os nós auxiliares. O custo total do CARCC2h é de  $O(\Delta(G)^3 + \Delta(G)^3 \times \log \Delta(G))$  em termos de tempo.

O Algoritmo 6 considera apenas a criação de enlaces bidirecionais. Porém, a solução pode ser facilmente generalizada para a criação de enlaces direcionais independentes. As linhas 11-13 podem ser modificadas para selecionar o melhor par de nós tal que a origem de cada um pertença a um componente diferente. Ainda, é possível que um único enlace cooperativo seja criado. Esta solução seria viável, por exemplo, para criar um enlace cooperativo entre um conjunto de nós sensores isolados quando estes devem enviar os dados para o componente adjacente que contém o sorvedouro ou que tenha um caminho para ele. Neste caso, para determinar a direção do enlace, pode-se fazer uso da estrutura da árvore criada pelo DAS.

### 5.1.2 Resultados Empíricos

Para avaliar a técnica proposta e para permitir a replicação dos resultados, um simulador foi desenvolvido em Matlab<sup>®</sup>. O processo de simulação consiste em gerar uma topologia aleatória e executar os algoritmos para detecção de articulações e pontes. Em seguida, um elemento crítico é removido e o algoritmo proposto é executado para verificar como o sistema recupera a conectividade. O cenário de simulação é baseado em parâmetros adaptados de [36, 61, 62]. A densidade do grafo é variada de  $4 \times 10^{-4}$ ,  $4,5 \times 10^{-4}$ , ...,  $12 \times 10^{-4}$  em uma área de  $300m \times 300m$ . A simulação utiliza o modelo proposto na Seção 3.2. A potência máxima de transmissão foi definida em  $P_{MAX} = 4900$  unidades de potência e o limiar para a potência recebida é definida com  $\tau = 1$  permitindo um raio máximo de transmissão  $R_{MAX} = 70m$ .

Após a execução do algoritmo DAS, os grafos conectados que contém ao menos um elemento crítico são considerados. Em seguida, é introduzida uma falha em uma articulação escolhida aleatoriamente. O sistema proposto é aplicado para verificar sua habilidade em recuperar a conectividade. Sendo realizada por meio de duas opções: um enlace bidirecional ou dois enlaces direcionais independentes (um para cada direção). Os resultados apresentados são oriundos de 1.000 simulações, para cada densidade, com o nível de significância de 95% [48]. Para simplificar a apresentação dos dados, os intervalos de confiança

são demonstrados somente para o melhor e pior caso. As seguintes métricas são utilizadas para verificar o desempenho da solução proposta:

- Métrica  $M1$  ( $W_D(H_{i,j})$ ): potência de transmissão necessária para atingir o nó mais distante no conjunto de auxiliares;
- Métrica  $M2$  ( $W_{CC}(H_{i,j})$ ): a potência de transmissão necessária individualmente durante a cooperação - todos os nós participam com a mesma potência de transmissão;
- Métrica  $M3$  ( $|H_{i,j}|$ ): número de auxiliares necessários para manter a conectividade com o nó de destino escolhido;
- Métrica  $M4$  ( $W_D(H_{i,j}) + (|H_{i,j}| + 1)W_{CC}(H_{i,j})$ ): a potência de transmissão total necessária para estabelecer a conectividade utiliza a comunicação cooperativa - esta métrica compreende a potência empregada na fase de cooperação e na fase de coordenação;
- Métrica  $M5$  ( $W_D$ ): a potência de transmissão total necessária para estabelecer a conectividade por meio de enlaces diretos entre os nós mais próximos, isto é, a com uma potência além de  $P_{MAX}$ ;
- Métrica  $M6$ : número de chamada à rotina DBHSS para a seleção de auxiliares;

Para permitir a avaliação da solução proposta, foram computados os melhores enlaces cooperativos entre os componentes com informação global da topologia. Assim, para cada grafo, são simulados mais dois cenários: o melhor enlace bidirecional global e o melhor par de enlaces direcionais (um para cada direção). O melhor enlace cooperativo global é equivalente a criar todos os enlaces cooperativos possíveis e, em seguida, aplicar alguma técnica de CT, como o MST, para manter apenas os enlaces de melhor custo entre os componentes. Assim como em trabalhos correlatos, computar enlaces entre cada nó da topologia depende da posição geográfica [61, 62]. Esta informação pode ser obtida através do *Global Positioning System* (GPS), ou Sistema de Posicionamento Global. O enlace cooperativo bidirecional de menor custo é equivalente àquele criado pelo *CoopBridges* [61]. Para a apresentação dos resultados, o custo (potência de transmissão ou número de auxiliares) corresponde à média entre os valores de cada direção. Todas as métricas foram coletadas em função da quantidade de nós por área. Assim, o eixo “x” das figuras a seguir, representa a densidade. Para facilitar a apresentação dos dados, é adotada a seguinte nomenclatura:

- **Local**  $\leftrightarrow$ : um enlace bidirecional criado entre pares de componentes utilizando informação localizada a 2-saltos;

- **Global  $\leftrightarrow$** : um enlace bidirecional criado entre pares de componentes utilizando informação global da topologia;
- **Local  $\rightarrow \leftarrow$** : dois enlaces direcionais criados entre pares de componentes utilizando informação localizada a 2 saltos;
- **Global  $\rightarrow \leftarrow$** : dois enlaces direcionais criados entre pares de componentes, com informação global da topologia;
- **Melhor enlace direto local**: enlace direto entre os dois vizinhos da articulação mais próximos entre si, um em cada componente;
- **Melhor enlace direto global**: enlace direto entre os dois nós mais próximos na topologia global.

A Figura 5.4 apresenta os resultados da simulação para a Métrica  $M1$ . O eixo “y” corresponde à potência de transmissão empregada pela origem para enviar os dados na primeira fase da comunicação cooperativa. Nesta figura, a linha marcada com quadrados corresponde à potência aplicada pelo enlace cooperativo local bidirecional (Local  $\leftrightarrow$ ), a linha marcada com pontos corresponde a dois enlaces cooperativos direcionais locais (Local  $\rightarrow \leftarrow$ ), a linha marcada com estrelas corresponde ao melhor enlace cooperativo bidirecional global (Global  $\leftrightarrow$ ) e a linha marcada com triângulos corresponde ao melhor par de enlaces cooperativos direcionais globais (Global  $\rightarrow \leftarrow$ ). Como pode ser observado,

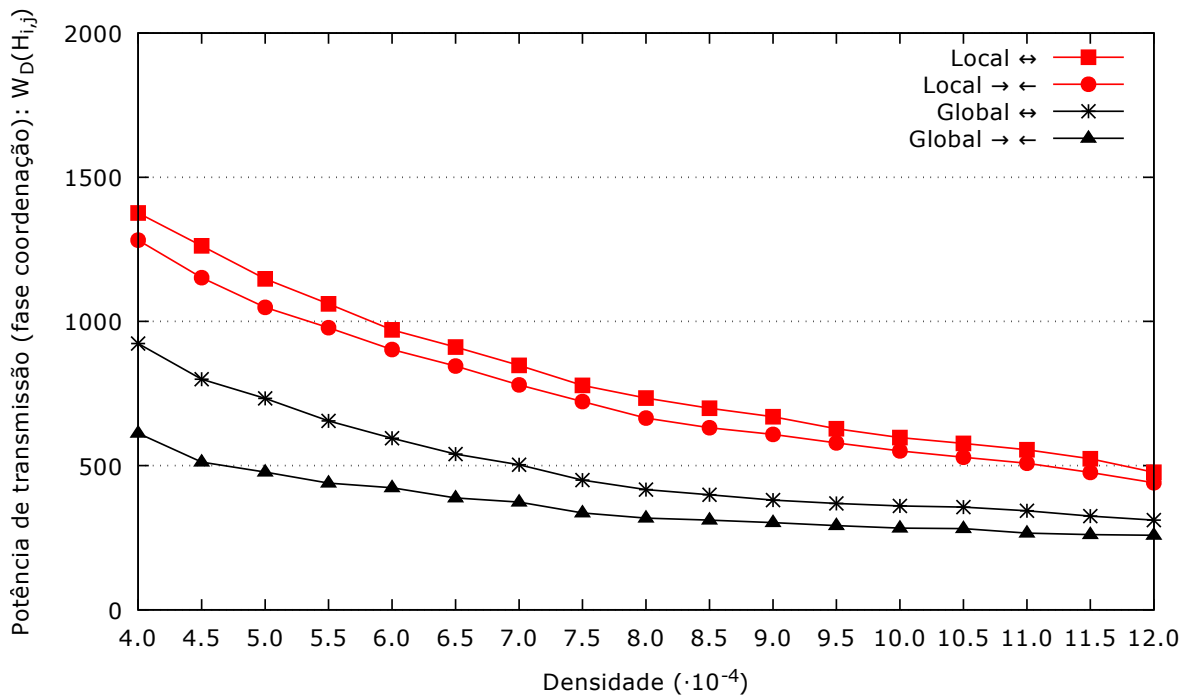


Figura 5.4: CARCC2h: Potência de transmissão para a fase de coordenação.

a potência necessária para alcançar o auxiliar mais distante reduz com o aumento na densidade. Com o incremento na densidade, a distância média entre os nós diminui e, conseqüentemente, com auxiliares mais próximos, o custo  $w_d(H_{i,j})$  também reduz. Os enlaces criados, por meio de informação localizada sobre a topologia, requerem uma potência de transmissão maior que os cenários em que se aplica informação global da topologia. Este comportamento resulta da possibilidade de encontrar, em qualquer região do grafo, uma tupla (origem, destino, auxiliares) com custo potencialmente mais baixo. Ao adotar informação localizada, os enlaces cooperativos são criados em uma região geográfica limitada às proximidades das articulações. No pior caso, com densidade de  $4 \times 10^{-4}$ , a solução localizada requer 1326,27 e 1281,7 unidades de potência da origem para comunicar com os auxiliares, respectivamente, para enlaces bidirecionais e dois enlaces direcionais. Utilizando informação global, foram necessárias 923,36 e 612,57 unidades de potência, nesta ordem, para os mesmos cenários. Em ambas alternativas, o maior intervalo de confiança é de  $\pm 125,52$ . Em média, os enlaces bidirecionais requerem 812,50 unidade de potência para os cenários em que enlaces bidirecionais são criados com informações locais e e 498,15 ao fazer uso de informação global. Quando dois enlaces direcionais são criados, precisou-se, em média, 747,04 unidades de potência ao empregar informação local e 360,60 ao aplicar informação global. No melhor caso, com densidade de  $12 \times 10^{-4}$ , a solução localizada requer 477,05 e 440,35 unidades de potência, respectivamente, para enlaces bidirecionais e dois enlaces direcionais. Utilizando informação global, a comunicação com os auxiliares requer, nesta ordem, 311,11 e 257,65, para os mesmos cenários. Para o melhor caso, o maior intervalo de confiança é de  $\pm 90,73$  unidades de potência. Observe que a criação de um enlace bidirecional corresponde à criação de dois enlaces direcionais em que o nó de destino em uma direção é, necessariamente, o nó de origem na outra direção. Assim, o número de opções para criar o enlace bidirecional é menor, acarretando em um custo maior. A diferença média quando se aplica informação global é de e 79,14%, no pior caso, e 61,79% no melhor caso. Com o aumento na densidade, a solução proposta tende a melhorar o seu desempenho.

A Figura 5.5 apresenta os resultados da simulação para a Métrica  $M2$ . O eixo “y” corresponde à potência de transmissão necessária em cada nó participante da fase de cooperação (transmissão dos dados para o destino). São apresentados os mesmos cenários da Figura 5.4 com a mesma configuração para cada linha. Como pode ser observado, a potência necessária nos nós participantes da segunda fase da Comunicação Cooperativa reduz com o aumento na densidade. Assim como ocorre com os nós, com o aumento na densidade, menor é a distância média entre os componentes. Além disso, a redução na distância média entre o destino e todos os nós participantes da fase de cooperação acarreta em aumento na participação de cada um em relação ao sinal recebido. Esta

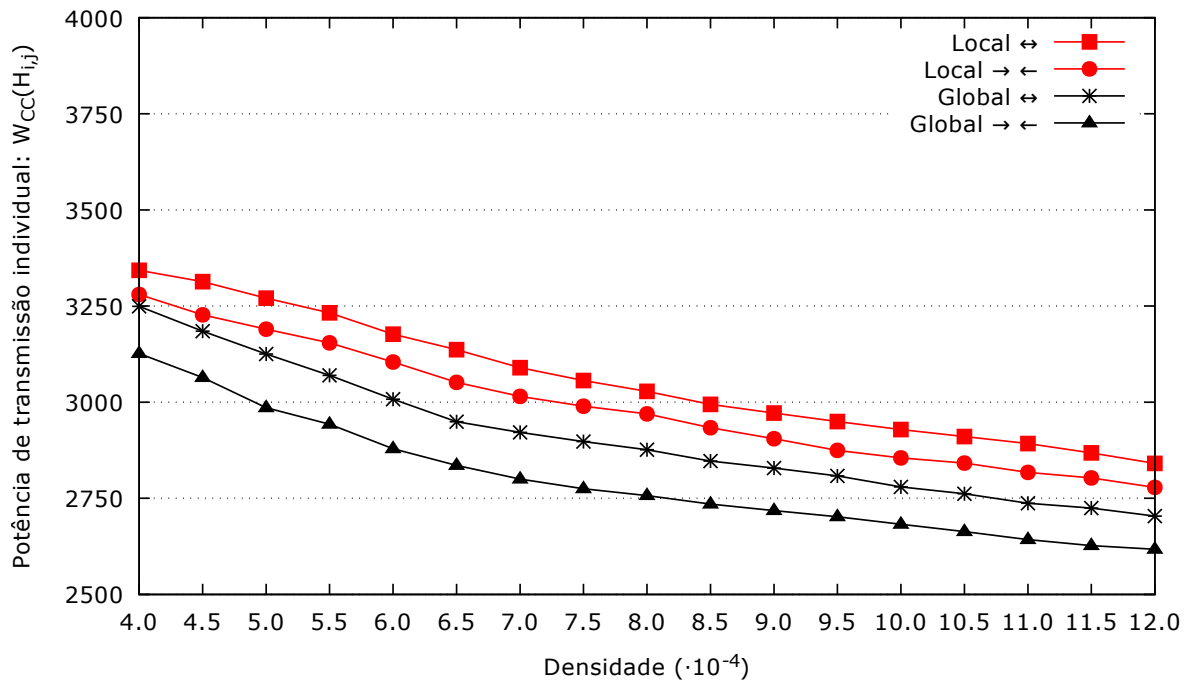


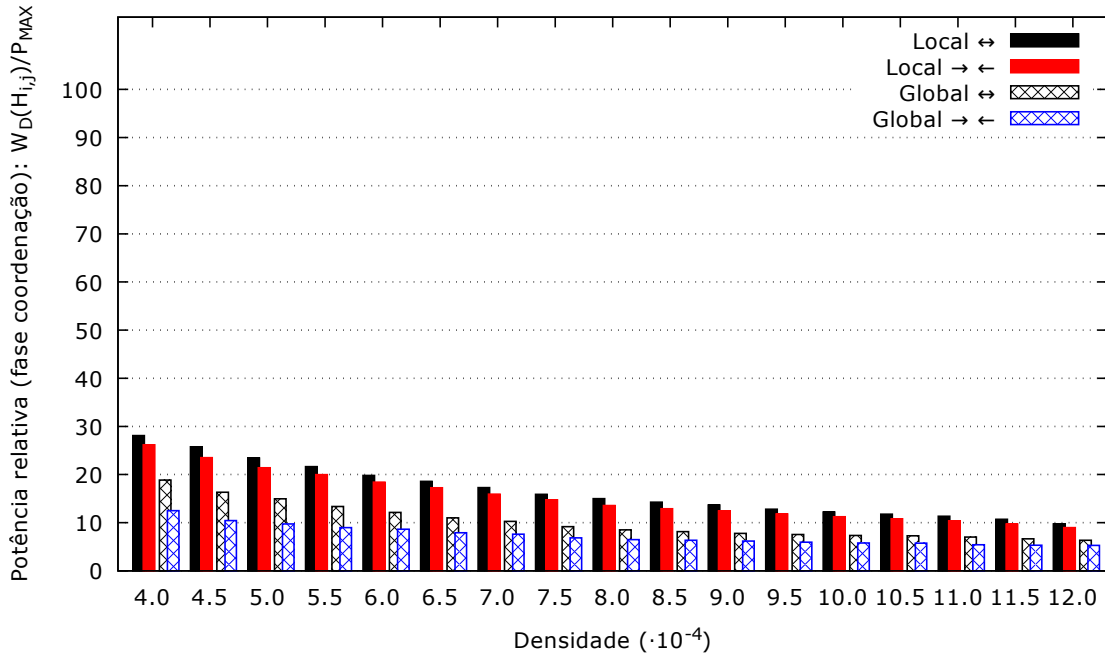
Figura 5.5: CARCC2h: Potência de transmissão para a fase de cooperação.

figura deve ser analisada em conjunto com a quantidade de auxiliares necessários durante a transmissão (Figura 5.7). Como existe uma pequena variação no número de auxiliares, é reforçada a hipótese de que os resultados apresentados correspondem à redução média na distância entre os componentes. No pior caso, com densidade de  $4 \times 10^{-4}$ , a solução localizada requer 3328,22 e 3280,36 unidades de potência para cada nó participante da fase de cooperação, respectivamente, para enlaces bidirecionais ou dois enlaces direcionais. Utilizando informação global, cada nó requer, para estes mesmos casos, 3249,43 e 3126,81 unidades de potência. O maior intervalo de confiança é de  $\pm 61,79$ . Na média, os cenários em que enlaces bidirecionais são criados, foi necessário 3059,13 unidade de potência usando informação local e 2910,18 utilizando informação global. Quando dois enlaces direcionais são criados, foi necessário, em média, 2987,73 unidades de potência aplicando informação local e 2797,31 adotando informação global. No melhor caso, com densidade de  $12 \times 10^{-4}$ , a solução localizada requer 2841,05 unidades de potência para o enlace bidirecional e 2778,21 para dois enlaces direcionais. Na informação global, os nós utilizaram, para estes mesmos casos, 2703,76 e 2617,44 unidades de potência. O maior intervalo de confiança, para o melhor cenário é de  $\pm 58,41$  unidades de potência. Em geral, os enlaces locais exigem 5,96% mais potência de transmissão individual que os enlaces globais. Em média, a solução local requer que os nós participantes utilizem 61,70% da potência máxima de transmissão dos nós, ao passo que, ao aplicar informação global da topologia, os nós necessitam de 59,95%. Estes dados indicam que a solução localizada consegue recuperar

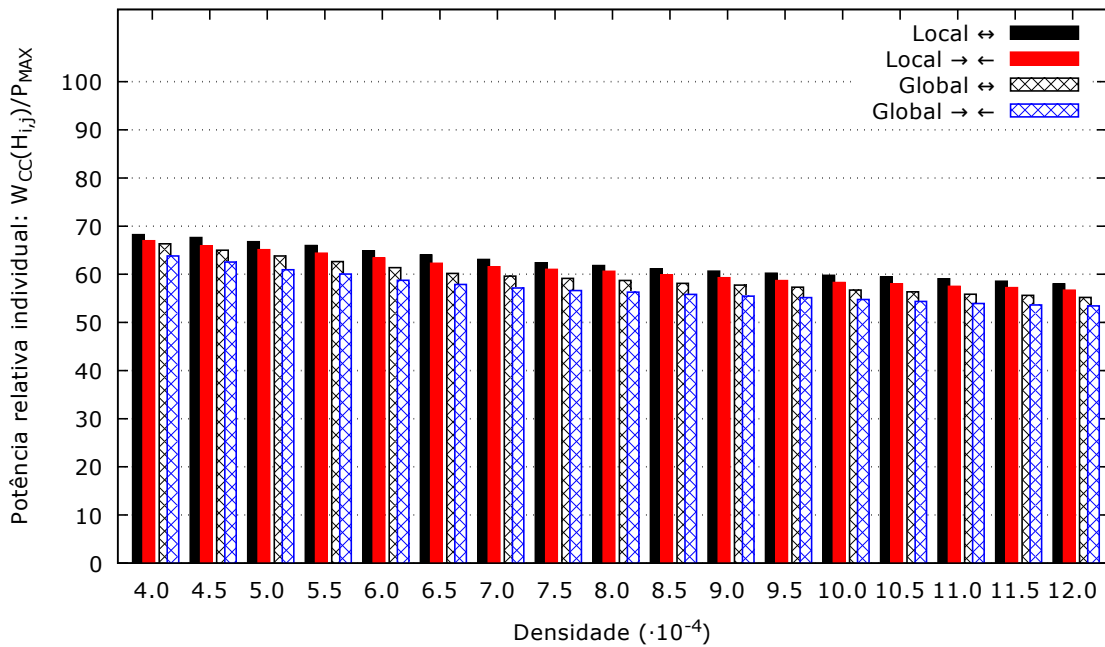


a conectividade empregando uma potência de transmissão similar à necessária quando se aplica o melhor enlace cooperativo criado por meio da informação global da topologia.

A Figura 5.6 apresenta, no eixo “y”, a relação entre a potência de transmissão utilizada e a potência de transmissão máxima. As barras sólidas representam a potência



(a)



(b)

Figura 5.6: CARCC2h: Potência de transmissão relativa.

utilizada pela solução proposta para um enlace bidirecionais ou dois enlaces direcionais, respectivamente nas cores vermelha e preta. As barras hachuradas de cor preta e azul representam, respectivamente, a potência necessária para o enlaces bidirecional ou o par de enlaces direcionais criados por meio da informação global da topologia. A Figura 5.6(a) apresenta a porcentagem da potência máxima de transmissão necessária para atingir o nó auxiliar mais distante. A potência de transmissão média é de 16,60%, 15,25%, 10,16% e 7,37%, respectivamente, para o enlace localizado bidirecional, os dois enlaces localizado direcionais, o enlace global bidirecional os dois enlaces globais direcionais. A potência de transmissão necessária quando se utiliza informação localizada se aproxima dos cenários em que se utiliza informação global da topologia com o incremento na densidade. Estes dados também demonstram que, em geral, os vizinhos das articulações conseguem selecionar vizinhos que requerem no máximo 7,88% mais energia que os nós selecionados para criar os enlaces cooperativos através de informação global da topologia. A Figura 5.6(b) apresenta a porcentagem da potência máxima de transmissão utilizada pelas nós participantes da fase de cooperação. A potência de transmissão média é de 62,43%, 60,97%, 59,39% e 59,39%, respectivamente, para o enlace localizado bidirecional, os dois enlaces localizado direcionais, o enlace global bidirecional os dois enlaces globais direcionais. A diferença média entre os enlaces localizados e os globais é de 3,46%, enquanto a maior diferença é de apenas 5,34%. Durante a CC em duas fases, a origem transmite os dados duas vezes. Porém, ao observar todas as soluções em ambas as figuras, podemos notar que a origem utiliza um montante da potência de transmissão inferior a potência máxima de transmissão. Isto indica que a origem consegue comunicar, por meio do enlace cooperativo, utilizando uma quantidade de energia inferior a aquela necessária quando se aplica a potência máxima de transmissão. Os nós auxiliares utilizam no máximo 62,43% da potência máxima de transmissão.

A Figura 5.7 mostra os resultados da simulação para a Métrica  $M3$ . O eixo “y” corresponde à quantidade de auxiliares que participam da fase de cooperação. São apresentados os mesmos cenários da Figura 5.4 com a mesma configuração para cada linha. Os resultados correspondem à média do número de auxiliares necessários para cada direção. Como pode ser observado, o número médio de auxiliares é, aproximadamente, 1 para todos os cenários. A diferença ao utilizar informação global e local é de 0,2%, enquanto o maior intervalo de confiança, para todos os cenários é de  $\pm 0,028$  nó. Estes dados indicam que, em geral, são necessários apenas dois nós transmitindo, simultaneamente, durante a fase de cooperação, para permitir a comunicação entre os componentes de rede. Para explicar esta quantidade de auxiliares e sua pequena variação, observe a distância entre os componentes. Quando conectados por uma articulação que não forma ponte, a distância máxima é de, no máximo,  $2 \times R_{MAX}$ , na medida em que a maior distância é

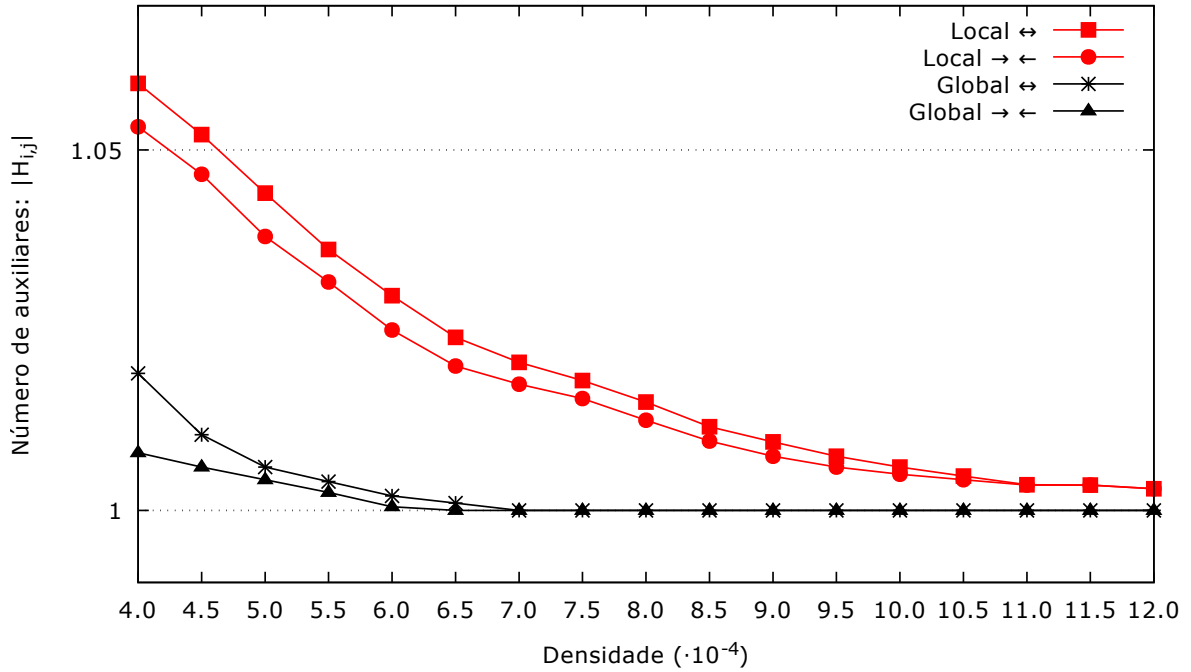


Figura 5.7: CARCC2h: Número de auxiliares.

de  $3 \times R_{MAX}$  quando ligados por uma ponte. Assim, apesar da distância média entre os nós (e entre os componentes) depender da densidade [28, 37], a distância máxima entre os componentes biconectados ligados por pontes e articulações tem um limite máximo, que pode ser definido pelo raio de transmissão máximo  $R_{MAX}$ . Outro fato determinante é o comportamento da heurística de seleção de auxiliares. Sua execução termina, durante a adição dos candidatos ao conjunto de auxiliares, assim que for possível estabelecer o enlace cooperativo. É utilizado o número mínimo de auxiliares. Observando sobre este aspecto, para a recuperação de conectividade por falhas em elementos críticos, o custo computacional médio do DBHSS é menor que o limite superior. O custo computacional médio é de  $\Theta(\Delta(G) \log \Delta(G))$  tempo, considerando que, em geral, somente um auxiliar é necessário.

A Figura 5.8 apresenta os resultados de simulação para a Métrica  $M4$ . O eixo “y” corresponde à soma das potências da fase de coordenação (comunicação com o auxiliar mais distante) e da fase de cooperação (soma da potência utilizada pela origem e pelos auxiliares ao transmitir os dados para o destino). São apresentados os mesmos cenários da Figura 5.4 com a mesma configuração para cada linha. A potência necessária corresponde à média para cada uma das direções. A potência total reduz com o aumento na densidade, como resultado na redução na potência necessária durante as etapas de coordenação e cooperação. Assim como esperado, quando a informação global da topologia for utilizada, é possível encontrar enlaces cooperativos com menor custo. No pior caso, com densidade

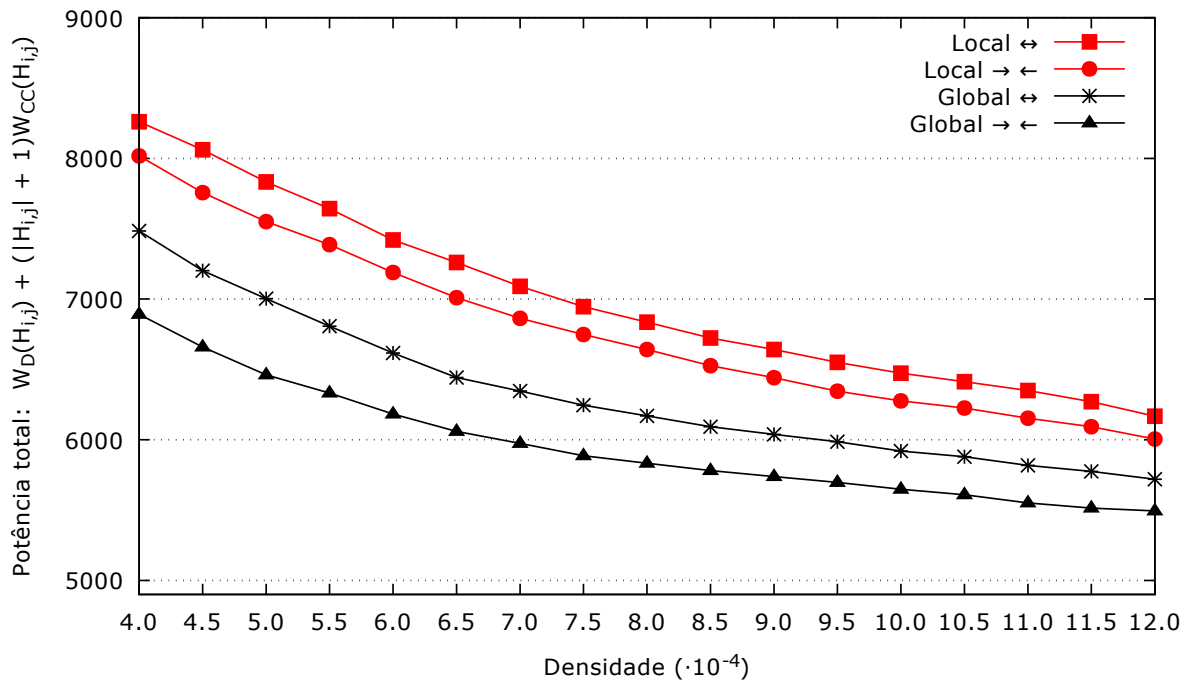


Figura 5.8: CARCC2h: Potência de transmissão total.

de  $4 \times 10^{-4}$ , a solução localizada requer 8260,69 e 8016,91 unidades de potência para restabelecer a conectividade, respectivamente, para enlaces bidirecionais ou dois enlaces direcionais. Com a informação global, a conectividade foi estabelecida, para estes mesmos cenários, aplicando 7483,97 e 6891,2 unidades de potência. O maior intervalo de confiança é de  $\pm 267,30$ . Na média, os cenários em que enlaces bidirecionais são criados, foi necessário 6996,30 unidade de potência usando informação local e 6325,95 para informação global. Quando dois enlaces direcionais são criados, foi necessário, em média, 6777,98 unidades de potência para informação local e 5959,35 para informação global. No melhor caso, com densidade de  $12 \times 10^{-4}$ , a solução localizada requer 6167,68 e 6005,13, respectivamente, para enlaces bidirecionais e dois enlaces direcionais. fazendo uso da informação global, os nós aplicaram, para estes mesmos casos, 5718,66 e 5493,54. O maior intervalo de confiança, para o melhor cenário é de  $\pm 121,058$  unidades de potência. A potência de transmissão necessária, quando se utiliza informações localizadas é, em média, 12,16% maior que aquela necessária quando se adota informação global. Esta diferença cai de 13,74% para 10,60% indicando que, com o aumento da densidade, a solução local tende a obter desempenho mais próximo ao cenários que empregam informação global.

A Figura 5.9 apresenta os resultados da simulação para a Métrica  $M5$ . O eixo “y” corresponde à potência de transmissão direta, quando o limite de potência  $P_{MAX}$  não é aplicado, para os nós mais próximos, considerando a topologia global e a topologia a 1 salto à partir da articulação. Também são apresentados os valores para a potência

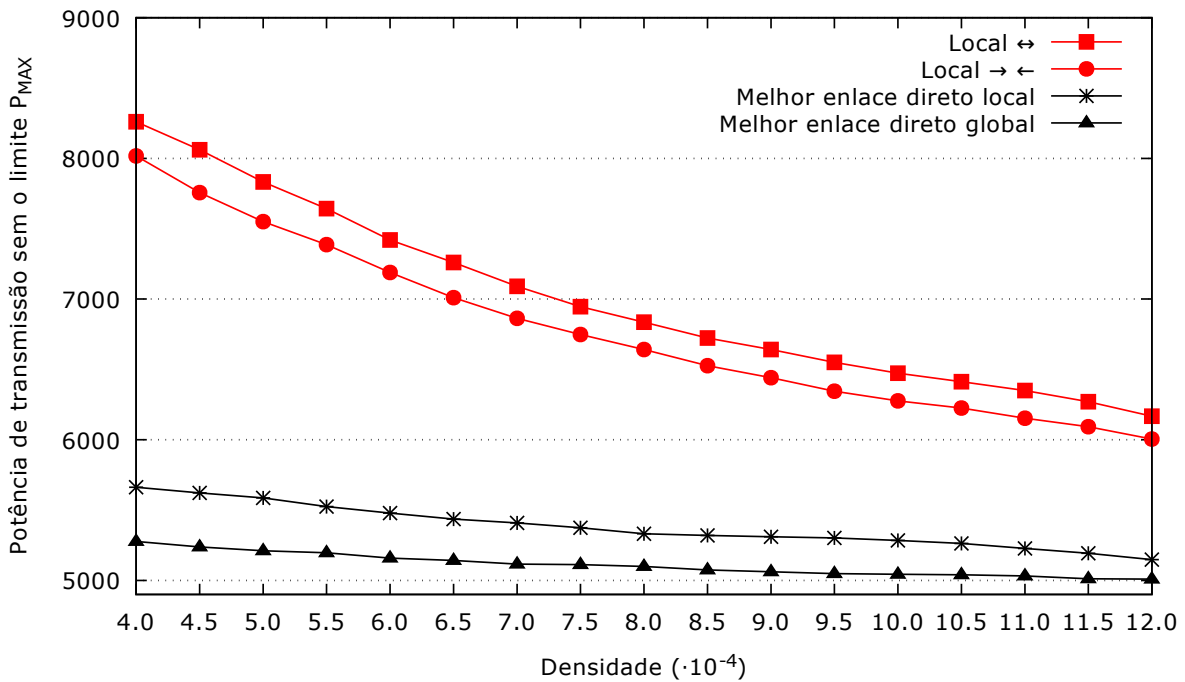


Figura 5.9: CARCC2h: Potência de transmissão sem o limite  $P_{MAX}$ .

de transmissão total requerida pelos enlaces cooperativos localizados. O uso dos enlaces diretos é uma situação improvável já que não é possível aumentar a potência de transmissão além do limite  $P_{MAX}$ . Por outro lado, estes valores permitem estabelecer um limite inferior para a potência de transmissão necessária para reconectar os componentes de rede. Nesta figura, a linha marcada com quadrados se refere à potência usada pelo enlace cooperativo local bidirecional (Local  $\leftrightarrow$ ), a linha marcada com pontos corresponde a dois enlaces cooperativos direcionais locais (Local  $\rightarrow \leftarrow$ ), a linha marcada com estrelas corresponde à potência de transmissão direta entre os nós mais próximos na topologia localizada e a linha marcada com triângulos demonstra a potência de transmissão direta entre os nós mais próximos considerando a topologia global. A diferença na potência de transmissão necessária pelo enlace direto global é menor que a requerida pelo enlace obtido com informação local. Este comportamento indica que é possível encontrar pares de nós entre os componentes que são mais próximos que os vizinhos dos elementos críticos a 1 salto. No pior caso, para a densidade de  $4 \times 10^{-4}$ , a solução proposta necessita 48,91% mais potência que os enlaces diretos. No melhor caso, a solução local exige 19,94% unidade de potência a mais. Em média, a solução proposta requer um montante de 31,34% a mais de potência de transmissão. Pode-se notar que a solução localizada se aproxima do limite inferior quando a densidade aumenta.

A Figura 5.10 apresenta os resultados da simulação para a Métrica  $M6$ . O eixo “y” representa a quantidade de vezes que a heurística DBHSS foi utilizada. A linha marcada

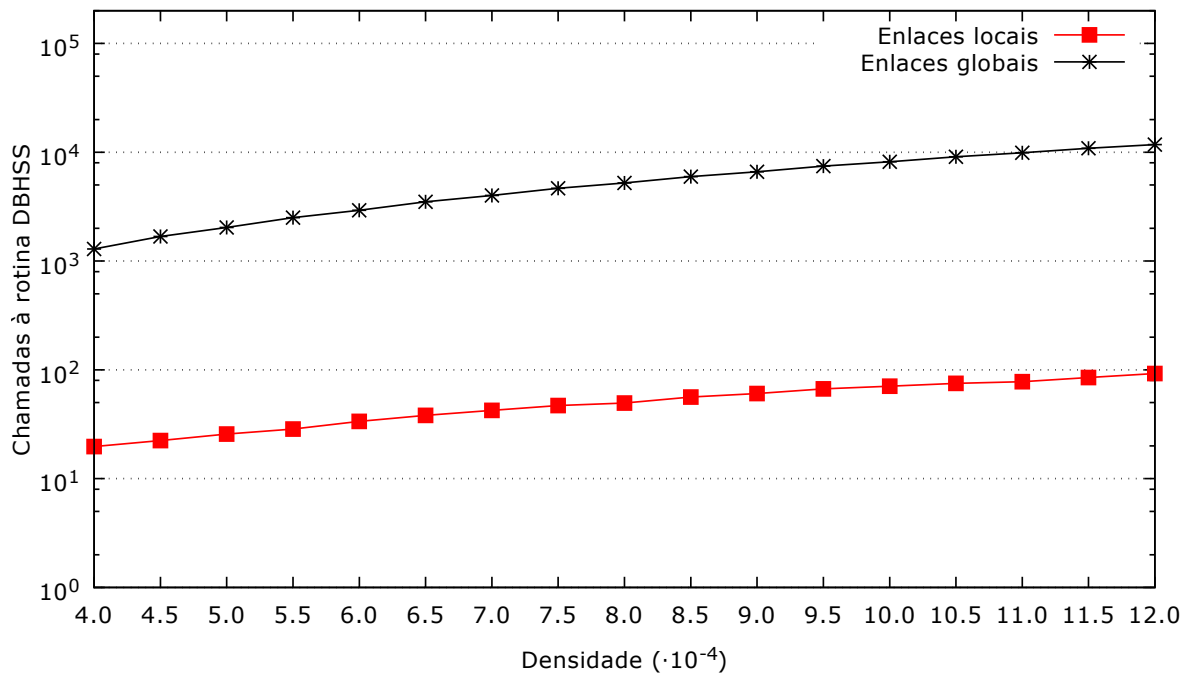


Figura 5.10: CARCC2h: Custo computacional em relação do uso do DBHSS.

com quadrado corresponde ao número de chamada à rotina para criar os enlaces locais, já a de estrela se refere aos globais. Com o aumento da densidade, maior é o número de vezes em que a rotina foi aplicada. Este comportamento equivale ao aumento no número de nós da rede e no número de vizinhos de cada nó. Nos cenários avaliados, para criar os enlaces cooperativos adotando informação global da topologia, a rotina DBHSS foi utilizada em um número superior a  $10^3$ . Com conhecimento localizado, o emprego da rotina foi inferior a  $10^2$ . Note que, para criar enlaces cooperativos por meio de informação global, a rotina deve ser utilizada um total de  $2 \times |V|^2$  vezes. Por meio de informação localizada, a rotina será aplicada apenas  $2 \times \Delta(G)^2$  vezes.

A Tabela 5.1 apresenta um resumo dos resultados obtidos pelo CARCC2h. A coluna “Potência individual” apresenta a relação entre a potência utilizada, nas duas fases da comunicação cooperativa, em relação à potência máxima de transmissão. A coluna “Potência do enlace direto” traz a relação entre a potência requerida pelos enlaces localizados e o limite inferior estabelecido por meio da potência de transmissão direta entre os nós mais próximos na topologia localizada. O número de opções se torna mais restrito quando são fixadas as origens e os destinos. Neste sentido, podemos notar que a potência individual requerida pelos nós é maior quando se empregam enlaces bidirecionais. Esta tabela também demonstra que a solução proposta requer um incremento de, no máximo, 30,08% em relação à potência utilizada pelo enlace diretor localizado quando a potência máxima de transmissão é ignorada. Apesar deste aumento, a potência individual máxima é de

62,43% em relação a  $P_{MAX}$ . Estes resultados reforçam que é viável usar a informação localizada para criar enlaces cooperativos para a recuperação de conectividade em caso de falhas em elementos críticos.

Tabela 5.1: CARCC2h: Relação da potência de transmissão utilizada.

	Potência individual ( $\times P_{MAX}$ )		Potência do enlace direto
	Fase coordenação	Fase cooperação	
Local bidirecional	16,59%	62,43%	130,08%
Local 2 direcionais	15,25%	61,07%	125,98%

O *CARCC2h* requer a coleta de informações localizadas a 2 saltos, a partir das articulações. Os nós críticos devem realizar os cálculos e transmitir duas mensagens para notificar os vizinhos selecionados para habilitá-los a recuperação de conectividade. Como alternativa, os vizinhos das articulações, designados como responsáveis por manter os enlaces cooperativos, podem realizar os cálculos. Ainda, os enlaces podem ser disponibilizados antes mesmo da segmentação da rede para ampliar os componentes biconectados ou realizar o balanceamento de carga entre eles. A seção a seguir apresenta uma solução que considera este cenário e não requer a troca de informações sobre a topologia da rede para criar os enlaces sobressalentes.

## 5.2 Solução Distribuída e Localizada a 1 Salto

Esta seção apresenta o *Connectivity Auto-Recovering through Cooperative Communication* (CARCC) ou Recuperação Automática de Conectividade por meio da Comunicação Cooperativa Utilizando Informações a 1 Salto, uma alternativa distribuída para a recuperação da conectividade com informação localizada a 1 salto. São três as principais diferenças para o CARCC2h: (i) o cálculo dos auxiliares e enlaces cooperativos é realizado pelos vizinhos das articulações; (ii) são necessárias apenas informações a um salto; (iii) a distância entre os componentes é estimada a partir de um valor arbitrário fornecido pela articulação. Em especial, o uso de valores arbitrários para a distância entre os componentes implica que os nós devem calcular se podem criar enlaces para atingir uma determinada distância e não um nó específico.

A ideia geral da solução consiste em determinar quais são os nós de articulação e, em seguida, estes nós devem notificar seus vizinhos para criar enlaces cooperativos sobressalentes para atingir outros componentes. Ao notificar seus vizinhos, a articulação fornece uma distância arbitrária entre ela e o componente mais distante. Assim, os nós de origem (vizinhos das articulações) podem determinar qual a potência necessária para atingir todos os componentes. A estimativa de distância pode ser muito além da distância

real, permitindo a criação de múltiplos enlaces. Assim, a solução compreende o ajuste da potência de transmissão. A subseção a seguir apresenta o CARCC em detalhes.

### 5.2.1 Detalhamento da Solução

Como esta solução se baseia apenas na topologia a um salto, a heurística de seleção de nós não dispõe das informações necessárias para determinar a potência recebida pelo destino. Assim, é necessária uma alternativa para seleção de auxiliares que: (i) utiliza um valor arbitrariamente estimado para determinar a distância entre a origem e o destino e, entre os auxiliares e o destino; (ii) calcula a potência de transmissão recebida pelo destino considerando que este é um nó posicionado arbitrariamente na rede a uma distância baseada na estimativa fornecida; (iii) não utiliza a distância até o destino como métrica ao adicionar os candidatos ao conjunto de auxiliares. Em especial, no item (ii), não é realizada nenhuma inferência sobre a posição relativa do nó de destino arbitrário como o ângulo em que se situa em relação a origem e os auxiliares.

Para calcular a distância máxima entre a articulação e os componentes biconectados ligados a ela, considere  $v_a$  como o nó de articulação. Então, de acordo com [56],  $N(v_a)$  compreende os nós localizados em  $l \geq 2$  componentes distintos  $C_1, C_2, \dots, C_l$ . Os vizinhos da articulação,  $v_i$  e  $v_j$ , em componentes diferentes estão separados por uma distância  $R_{MAX} < d_{i,j} \leq 2 \times R_{MAX}$ , onde  $R_{MAX}$  denota a distância máxima de transmissão conforme a Equação 2.2. A Equação 3.3 permite calcular a potência necessária para a comunicação cooperativa entre um nó de origem  $v_i$  e um nó de destino  $v_j$ . O enlace  $\widehat{v_i v_j}$  é criado por  $v_i$  se o conjunto de auxiliares  $H_{i,j} \subseteq N(v_i)$ . Contudo, ao calcular a Equação 3.3, a distância  $d_{h,j}$ ,  $h \in H_{i,j}$ , é necessária. Esta informação não está disponível ao utilizar informação topológica a 1 salto. Obviamente, o nó  $h \in H_{i,j}$  pode assumir a distância  $3 \times R_{MAX}$  para alcançar  $v_j$  ao calcular Equação 3.3. Esta estimativa impõe o uso de uma potência maior que a necessária pela origem e seus auxiliares. Portanto, uma estimativa de distância  $\mathcal{E}_{h,j}$  é utilizada. Considere  $v_i$  e  $v_j$  como nós adjacentes da articulação  $v_a$  em diferentes componentes tal que  $N(v_i) \cap N(v_j) = v_a$ . Pelo modelo considerado neste trabalho,  $v_i$  conhece  $d_{i,h}$ ,  $\forall h \in N(v_i)$ . De forma semelhante,  $v_a$  também pode determinar  $d_{a,i}$  e  $d_{a,j}$ . Ao trocar estas informações de distância o nó  $v_h$  pode calcular a estimativa de distância  $\mathcal{E}_{h,j}$  da seguinte forma:

$$\mathcal{E}_{h,j} = d_{h,i} + d_{i,a} + d_{a,j}. \quad (5.3)$$



---

**Algorithm 7** EDBHSS( $v_i, v_a, d_{a,j}$ )

---

**Require:** Source node  $v_i$ , articulation node  $v_a$ , estimated distance  $d_{a,j}$  ;

**Ensure:** The smaller helper set  $H_{i,j}$  complying with Eq. (3.3) and the required transmission power  $W_{CC}(H_{i,j})$  to create a cooperative link from  $v_i$  to  $v_j$ ;

# Let  $v_j$  be an arbitrary destination node such that  $v_j \in N(a)$  with  $d_{a,j} \leq r$ ;

1:  $\forall i \in N(v_i), W_i = d_{v_i,i}$ ;

2:  $C \leftarrow N(v_i)$  sorted by  $W_i$ ;

3:  $H_{i,j} \leftarrow \emptyset$ ;

4: **repeat**

5:   Remove first node from  $C$  and add to  $H_{i,j}$

6:   Node  $v_i$  compute  $P^r$  of  $v_i v_j$  and  $W_{CC}(H_{i,j})$  using Equations 3.3, 5.3 and 5.4;

7: **until**  $P^r < \tau$  and  $C \neq \emptyset$ ;

8: **if**  $P^r \geq \tau$  **then**

9:   return  $H_{i,j}$  and required transmission power;

10: **else**

11:   return  $H_{s,d} = \emptyset$  and  $W_{CC}(H_{i,j}) = 0$ ;

12: **end if**

---

Assim,  $R_{MAX} < \mathcal{E}_{h,j} \leq 3 \times R_{MAX}$ . De forma análoga, a distância arbitrária entre a origem e o suposto destino pode ser calculada por:

$$\mathcal{E}_{i,j} = d_{i,a} + d_{a,j}. \quad (5.4)$$

Neste caso,  $R_{MAX} < \mathcal{E}_{i,j} \leq 2 \times R_{MAX}$ . Estas estimativas podem ser utilizadas pela heurística de seleção de nós para calcular a Equação 3.3 baseado em informação a 1 salto. Com essa nova estratégia, o enlace cooperativo  $\widetilde{v_i v_j}$  conecta  $v_i$  a um nó arbitrário  $v_j$ , onde  $v_j$  reside em uma distância estimada  $\mathcal{E}_{i,j}$  de  $v_i$  e a distância  $v_h \in H_{i,j}$  é determinado por  $\mathcal{E}_{h,j}$ . A heurística *Estimated Distance Based Helper Set Selection* (EDBHSS) ou Seleção de Auxiliares Baseada na Distância Estimada, que utiliza esta estratégia, é apresentada em detalhes no Algoritmo 7. Em seguida, a heurística é explicada em detalhes.

O EDBHSS recebe como parâmetro o nó de origem  $v_i$ , a articulação  $v_a$  e a estimativa de distância  $d_{a,j}$  entre a articulação e os componentes biconectados. As principais diferenças em relação ao DBHSS são: (i) não é necessário especificar o nó de destino; e (ii) ao ordenar os candidatos a auxiliares, a distância entre eles e o destino não é utilizada (linhas 1-2). O restante do algoritmo é semelhante ao DBHSS, exceto pela linha 6 em que se utilizam as estimativas  $\mathcal{E}_{i,j}$  e  $\mathcal{E}_{h,j}$  para determinar a potência recebida pelo destino.

A recuperação de conectividade é realizada pelo CARCC. A solução tem como requisitos a identificação dos nós de articulação e a definição dos componentes biconectados da rede. Estas informações podem ser obtidas por meio das variáveis locais mantidas pelos algoritmos DAS e DDBC. Em função da interdependência, os algoritmos DAS, DBS e DDBC devem ser previamente executados nesta ordem. A recuperação da conectividade é realizada em quatro fases e tem como gatilho a definição de um nó como articulação. As seguintes tarefas são realizadas em cada fase:

---

**Algorithm 8** CARCC

---

**Require:** Boolean variable  $articulation_{v_a}$  from DAS algorithm and the array variable  $component_{v_a}$  from DDBC algorithm;

**Ensure:**  $\forall H_{i,j} \neq \emptyset$ , a CC-link  $\widetilde{v_i v_j}$  connecting  $\{v_i, v_j\} \in N(v_a)$  such that  $v_i \in C_k$  and  $v_j \in C_m$ ,  $k \neq m$ ;

# Computing the minimum distance to reach the farthest component

- 1:  $\forall u \in C_m$ ,  $1 \leq m \leq l$ ,  $v_a$  computes  $D_m = \min(d_{a,u})$ ;
- 2:  $v_a$  retrieves  $D_1$  and  $D_2$  satisfying  $D_1 \geq D_2 \geq \dots \geq D_m$ ;
- 3:  $v_a$  broadcasts  $(D_1, D_2)$  to all  $v \in N(v_a)$ ;

# Nodes receiving  $D_1, D_2$  verify whether a CC link can be established;

- 4:  $\forall v \in N(v_a)$  compute  $EDBHSS(v, v_a, D_1|D_2)$ ;
- 5: **if**  $H_{v,j} \neq \emptyset$  **then**
- 6:      $v$  sends  $\langle W_{CC}(H_{i,j}), v \rangle$  back to node  $v_a$ ;
- 7: **end if**

# Selecting a suitable backup node in each component  $C_m$ ;

- 8: Node  $v_a$  collects the messages and computes  $\min_{v_i \in C_m} (W_{CC}(H_{i,j}), v)$  for each component  $C_m$ ;
- 9: Node  $v_a$  informs to node  $v_i \in C_m$  that it holds de minimum cost among the nodes in that component. Node  $v_i$  becomes the backup node for component  $C_m$ ;

# Backup nodes adjust  $W_{CC}(H_{i,j})$  based on the received power  $P^r$ ;

- 10: The backup node in each component  $B_i$  sends a CC message  $M$  containing  $W_{CC}(H_{i,j}), SrcBlockID, SrcNode$ ;
  - 11: For each received message  $M$ , node  $v_z \in V$  records a tuple  $y$  with the  $(P_z^r, SrcBlockID, DestBlockID, SrcNode)$ ;
  - 12: Node  $v_z$  sends a message  $Y$  with all collected  $y$  back to the  $SrcNode$  of message  $M$  via node  $v_a$ ;
  - 13: For each component  $C_i$ , node  $v_a$  computes the  $P_{C_i}^r \leftarrow \max(P_z^r)$ ,  $z \in C_i$ ;
  - 14: Node  $v_a$  inform the  $P_{C_i}^r$  to  $SrcNode$  in component  $C_i$ ;
  - 15: On receiving the  $P_{C_i}^r$ , the  $SrcNode \in C_i$  adjust its transmission power  $W_{CC}^{opt}(H_{i,j}) \leftarrow \frac{W_{CC}(H_{i,j})}{P_{C_i}^r} \cdot \beta$ , ( $\beta > 0$ );
- 

1. Primeira fase (linhas 1-3): a articulação envia uma mensagem para seus vizinhos informando a estimativa de distância para os componentes adjacentes.
2. Segunda fase (linhas 4-7): cada nó vizinho da articulação deve, independentemente, identificar se são capazes de criar o enlace para a distância estimada.
3. Terceira fase (linhas 8-9): a articulação seleciona uma única origem baseado no custo informado.
4. Quarta fase (linhas 10-15): a origem selecionada utiliza uma mensagem especial para otimizar a potência de transmissão e reduzir a quantidade de enlaces criados.

O objetivo das fases 1 a 3 é obter uma origem com o menor custo para atingir a distância estimada, isto é:

$$\min(w_d(H_{i,j}) + (|H_{i,j}| + 1)w_{CC}(H_{i,j})), \quad (5.5)$$

enquanto a fase 4 obtém  $\min(w_{CC}(H_{i,j}))$  para manter a conectividade utilizando a origem e os auxiliares escolhidos nas fases anteriores. Os detalhes do CARCC são apresentados no Algoritmo 8. Na primeira fase (linhas 1-3), o CARCC calcula uma distância adequada para a criação dos enlaces CC que permita conexões entre os componentes de rede. Considere que  $l$  denota o número de componentes ligados pela articulação  $v_a$ , tal que  $C_1, C_2, \dots, C_l$ . Então, para cada bloco,  $C_i$ , ( $1 \leq i \leq l$ ), a articulação  $v_a$  calcula  $D_i = \min(d_{a,u})$ , onde  $u \in C_i$ . Esta é a distância mínima para alcançar o nó mais próximo no componente mais distante. Os dois maiores valores,  $D_1$  e  $D_2$ , são enviados para cada nó  $u \in N(v_a)$ . A segunda fase é executada nas linhas 5-7. Assim que a estimativa de distância está disponível para os vizinhos da articulação, eles executam a rotina *EDHSS* com a estimativa  $D_1$  ou  $D_2$ . Se  $d_{a,u} = D_1$ , então  $u \in N(v_a)$  utiliza  $D_2$  caso contrário,  $u$  aplica  $D_1$ . Isto é necessário já que  $u$  está a uma distância  $D_1$  e a estimada para alcançar o segundo nó mais distante em um componente é  $D_2$ . Estes nós tem informação suficiente para criar o enlace cooperativo para a distância fornecida. Assim eles enviam as respectivas estimativas de custo para  $v_a$ . Note que os nós  $u \in C_i \cap N(v_a)$  podem se abster de enviar o custo do enlace (linha 6) se, ocasionalmente, escutarem um custo inferior. Mesmo assim, a articulação pode ter vários candidatos por bloco  $C_i$ ,  $1 \leq i \leq l$ . A terceira fase permite selecionar o nó de origem que tem o menor custo. Esta tarefa é executada nas linhas 8-9. Neste estágio, cada bloco  $C_i$  tem um único nó de origem cujo custo é o menor entre os outros nós naquele componente. Considere  $v_i$  como o nó de origem selecionado no componente  $C_i$ . Como  $v_i$  usa uma distância estimada para calcular o enlace cooperativo, o custo pode demandar por mais potência que a necessária para alcançar os nós em diferentes componentes. Na quarta e última fase, os nós de origem  $v_i \in N(v_a)$ ,  $1 \leq i \leq l$ , ajustam sua potência de transmissão. Esta tarefa é executada nas linhas 10-15. A ideia é enviar uma mensagem  $M$  contendo  $W_{CC}(H_{i,j}), SrcBlockID, SrcNode$ . Todos os nós em componentes diferentes coletam  $M$  e registram a potência recebida ( $P_{SrcBlockID}^r$ ). Esta informação é encaminhada de volta para o nó  $v_i$  por meio de enlaces diretos usando uma mensagem  $Y$ . Como o nó  $v_a$  é uma articulação, todas as respostas devem passar por ela. Ao verificar cada mensagem, a articulação  $v_a$  pode identificar o maior  $P_{C_i}^r$  para cada par  $\widetilde{v_i v_j}$ , onde  $v_i \in C_i$  e  $v_j \in C_j$ ,  $i \neq j$ . O cálculo de  $P_{C_i}^r$  pode ser realizado nos nós intermediários, reduzindo o número de mensagens e a sobrecarga na articulação. Desta forma, a articulação deve receber, no máximo,  $\Delta(G)$  mensagens. Esta informação é então encaminhada para o nó  $v_i$ . Ao receber esta informação, o nó  $v_i$  pode ajustar sua potência de transmissão para alcançar o nó  $v_j \in C_j$ . O conjunto de auxiliares  $H_{i,j}$  não é modificado durante esta tarefa (a rotina *EDHSS* não é aplicada novamente). A potência de transmissão otimizada para alcançar  $v_j$  pode ser calculada como  $W_{CC}^{opt}(H_{i,j}) = \frac{W_{CC}(H_{i,j})}{P^r} \cdot \beta$ , onde  $\beta \geq 1$  é usado como margem de erro.

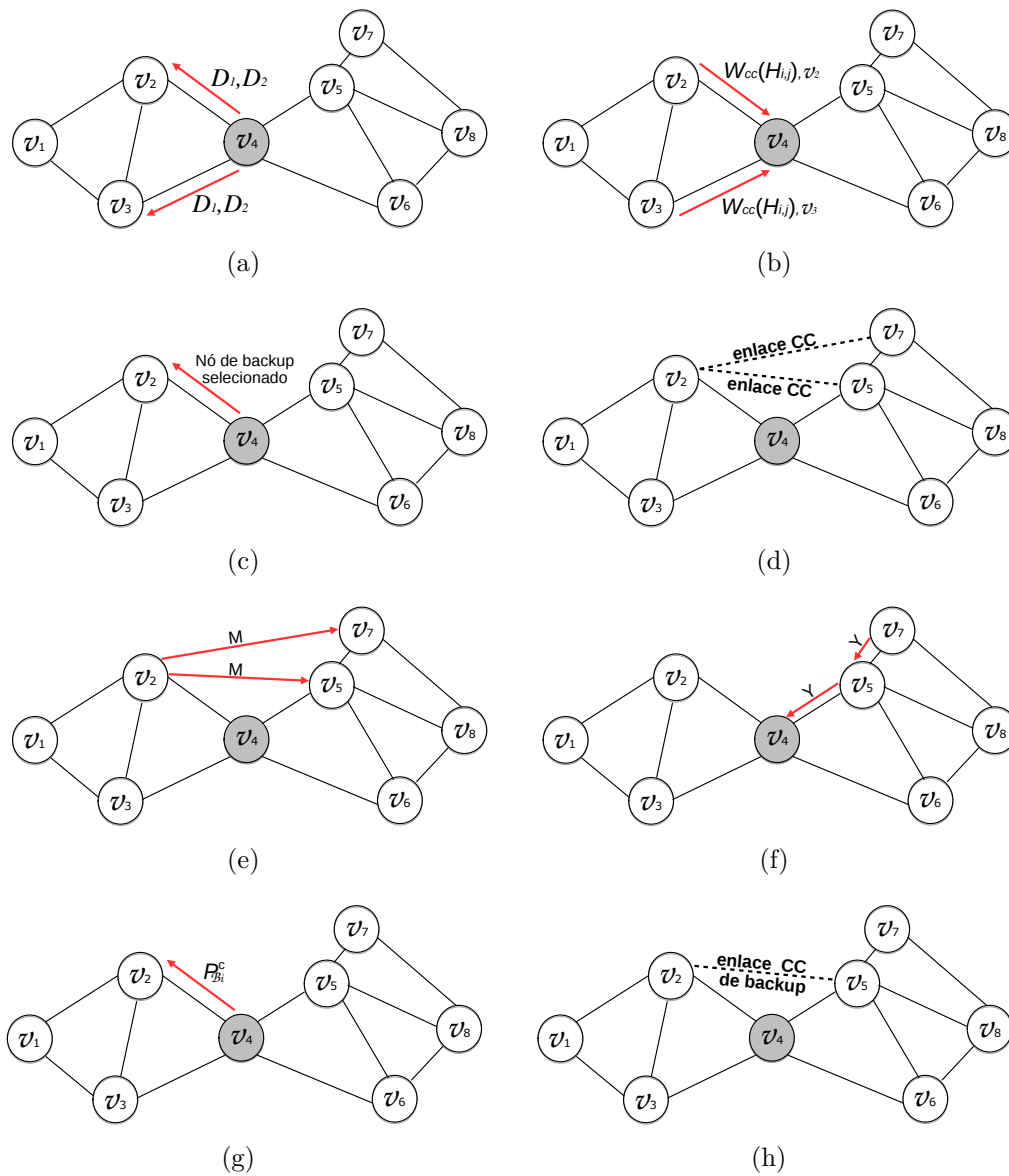


Figura 5.11: CARCC: Exemplo de execução do algoritmo.

A Figura 5.11 exemplifica, parcialmente, as tarefas do CARCC em um grafo composto por duas articulações e dois componentes de rede. Na Figura 5.11(a), a articulação calcula a distância estimada e envia para seus vizinhos de acordo com as linhas 1-3 do Algoritmo 8. Note que, apesar de não ilustrado, a estimativa de distância é enviada aos vizinhos no componente à direita. Na Figura 5.11(b), os nós verificam se a distância estimada pode ser alcançada (linhas 4-7). Se for possível, eles enviam o custo para a articulação. Na Figura 5.11(c), uma mensagem é usada pela articulação para definir o nó de origem (linhas 8-9). A Figura 5.11(d) mostra os enlaces cooperativos que podem ser criados por meio da distância sobrestimada e o nó de origem selecionado. Na Figura 5.11(e), a transmissão da mensagem  $M$  é usada para descobrir o  $P^r$  de cada nó que pode ser alcançado (linha

10). Na Figura 5.11(f), os nós que recebem  $M$  armazenam tuplas  $y$  e enviam a mensagem  $Y$  com elas para a articulação (linhas 11-12). Na Figura 5.11(g), a articulação seleciona  $P_{C_i}^r$  de maior valor e envia para o nó de origem (linhas 13-14). Na Figura 5.11(h) é apresentado o enlace sobressalente disponível após a otimização (linha 15). Note que, para este exemplo, foi criado apenas um enlace do bloco da esquerda para o da direita. O mesmo processo ocorre para criar o enlace da direita para a esquerda.

Para definir o custo computacional da solução proposta, considere que  $\Delta(G)$  é o grau do grafo. A rotina EDBHSS tem, essencialmente, o mesmo custo computacional que a rotina DBHSS e não utiliza nenhuma mensagem. Para o CARCC, o custo computacional pode ser definido em termos de tempo e número de mensagens. O algoritmo é distribuído e executa em todos os nós da rede. Seu gatilho é quando o nó determina que é uma articulação ou quando uma mensagem é recebida. Considere que  $l$  é o número de componentes ligados pela articulação  $v_a$ . A primeira fase (linhas 1-3) é executada na articulação  $v_a$  com o custo de  $O(\Delta(G))$ . A segunda (linhas 4-7) é executada em  $N(v_a)$ , com um custo total de  $O(\Delta(G) + \Delta(G) \log \Delta(G))$ , que corresponde ao uso da rotina EDBHSS na linha 4. A fase de seleção da origem é executada na articulação (linhas 8-9) e pode ser realizada em tempo  $O(\Delta(G))$ . A tarefa final (linhas 10-15) é executada na articulação e nos nós regulares e requer  $O(\Delta(G))$  tempo para calcular  $P_{C_i}^r$  e ajustar a potência de transmissão. Assim, o custo do CARCC é  $O(\Delta(G) \cdot \Lambda)$ , onde  $\Lambda$  corresponde ao custo para selecionar os nós auxiliares. O custo total do CARCC é  $O(\Delta(G)^2 + \Delta(G)^2 \log \Delta(G))$  em termos de tempo.

Para o custo de comunicação, observe que o CARCC envia mensagens nas linhas 3, 6, 9, 10 e 12. A solução proposta considera o uso da difusão da comunicação sem fio. Na primeira fase, a articulação envia uma única mensagem (linha 3) para propagar a estimativa de distância. Cada vizinho da articulação envia seu custo para criar o enlace cooperativo (linha 6) resultando em, no máximo,  $\Delta(G)$  transmissões. Para selecionar a melhor origem, são enviadas  $l$  mensagens (linha 9), uma para cada bloco. Para ajustar a potência total de transmissão, uma mensagem é enviada pelo nó selecionado usando a comunicação cooperativa em um total de  $l$  mensagens. Todos os nós que recebem a mensagem devem informar, a origem, qual a potência recebida e o componente a que pertence (linha 12). Este processo requer a transmissão de, no máximo,  $|V|$  mensagens considerando que os nós intermediários podem calcular o  $P_{C_i}^r$  quando encaminham a mensagem. Em resumo, o CARCC requer, no máximo,  $O(|V|)$  mensagens. Porém, como será mostrado na próxima seção, a quantidade de mensagens enviadas é, em média, menor que o limite, pois se restringe a uma posição geográfica próxima da articulação.

## 5.2.2 Resultados Empíricos

Para avaliar o desempenho do CARCC e permitir a replicação dos resultados, o algoritmo foi simulado no Matlab<sup>®</sup> com os mesmos parâmetros do CARCC2h. A densidade é variada de  $4 \times 10^{-4}$ ,  $4,5 \times 10^{-4}$ , ...,  $12 \times 10^{-4}$  em uma área de  $300m \times 300m$ . A simulação utiliza o modelo proposto na Seção 3.2. A potência máxima de transmissão foi definida em  $P_{MAX} = 4900$  unidades de potência e o limiar para a potência recebida é definida com  $\tau = 1$ , permitindo um raio máximo de transmissão  $R_{MAX} = 70m$ . O fator de ajuste de potência, usado como margem de erro, é definido como  $\beta = 1$ . As topologias aplicadas ao CARCC foram geradas de forma aleatória, cada uma contendo ao menos um elemento crítico. No total, 1.000 topologias distintas foram utilizadas. Para permitir a verificação da performance, também foram computados o melhor enlace cooperativo global e o melhor enlace cooperativo localizado a 2 saltos. Estes enlaces equivalem, respectivamente, à melhor solução exaustiva e ao CARCC2h quando apenas o enlace cooperativo direcional de menor custo é mantido. Para os cenários utilizados como referência, ao calcular o conjunto de auxiliares e a potência necessária com a heurística EDBHSS, aplicou-se a distância exata entre a origem e seus auxiliares, entre a origem e o destino e, entre os auxiliares e o destino.

A Figura 5.12 exemplifica a simulação do CARCC. Na Figura 5.12(a) é apresentado o grafo inicial. A Figura 5.12(b) apresenta o grafo resultante quando a articulação é removida do grafo. Note que o algoritmo deve ser executado antes da falha no elemento crítico. A Figura 5.12(c) traz o grafo com os enlaces criados pelo CARCC sem a otimização da potência de transmissão (linhas 1-7). Os enlaces cooperativos são desenhados em linha tracejada, enquanto as linhas pontilhadas representam os enlaces auxiliares (comunicação entre a origem e seus auxiliares). Note que cada vizinho da articulação pode criar múltiplos enlaces para o outro componente. A Figura 5.12(d) exhibe o grafo com o enlace cooperativo mantido pelo CARCC após o ajuste de potência. O enlace é criado antes mesmo da falha no elemento crítico, conforme esta figura.

Após a execução dos algoritmos DAS, DBS e DDBC, os grafos conectados que contém, ao menos um elemento crítico são considerados. O sistema proposto é aplicado para verificar sua habilidade em criar os enlaces sobressalentes. A recuperação é realizada por meio de enlaces direcionais. Os resultados apresentados são oriundos de 1.000 simulações, para cada densidade, com o nível de significância de 95%. As métricas  $M1$  a  $M5$ , utilizadas para avaliar o CARCC2h, são aplicadas na avaliação do CARCC. Nesta seção, a Métrica  $M6$  ( $|\tilde{E}|$ ) corresponde ao número de enlaces criados com a comunicação cooperativa.

A Figura 5.13 apresenta os resultados de simulação para a Métrica  $M1$ . O eixo “y” consiste na média da potência para a fase de coordenação (necessária para alcançar o auxiliar mais distante). A linha com triângulo se refere ao CARCC sem otimização (linhas

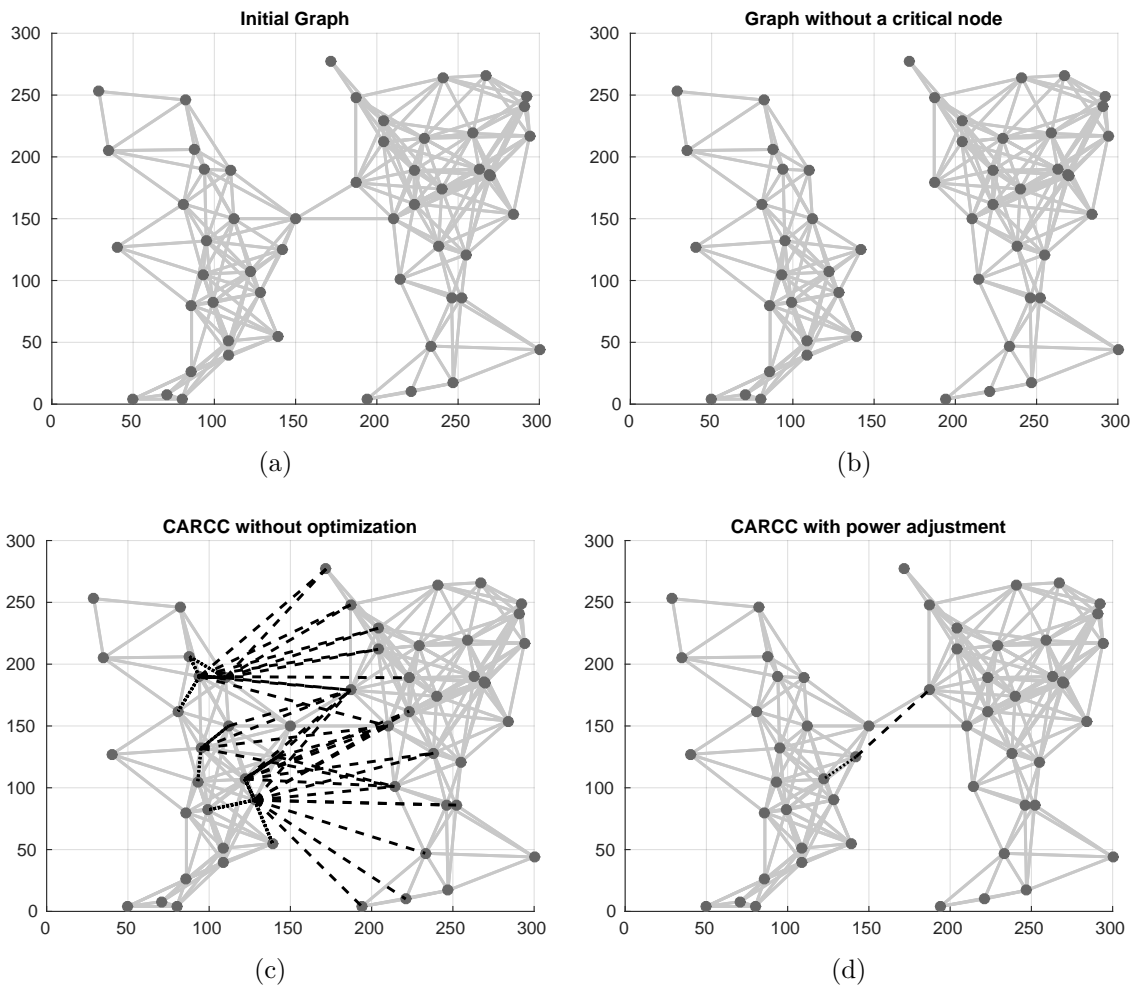


Figura 5.12: Exemplo de simulação do CARCC: (a) grafo inicial; (b) grafo resultante após a remoção de uma articulação; (c) enlaces cooperativos criados sem otimização (linhas 1-9); e (d) enlace cooperativo criado após a seleção da origem e ajuste de potência (linhas 1-15).

1-9 do algoritmo), a linha marcada por pontos corresponde ao CARCC após o ajuste da potência (linhas 1-15 do algoritmo), a linha com quadrado corresponde ao melhor enlace criado utilizando informação global da topologia e a linha marcada por estrela se refere ao melhor enlace criado por meio da informação localizada a 2 saltos. Assim como esperado, a potência de transmissão para alcançar o auxiliar mais distante reduz com o aumento na densidade. No pior caso, a potência requerida para atingir o auxiliar mais distante é de 384,92 unidades de potência para o melhor enlace global, 455,45 unidades para o melhor enlace local, 862,22 unidades para o CARCC sem otimização e 489,1 para o CARCC com ajuste de potência. O maior intervalo de confiança é de  $\pm 42,05$ . A média dos valores é 222,35 unidades para o melhor enlace global, 280,79 unidades para o melhor enlace local, 500,61 unidades para o CARCC sem otimização e 292,28 para o CARCC com ajuste de

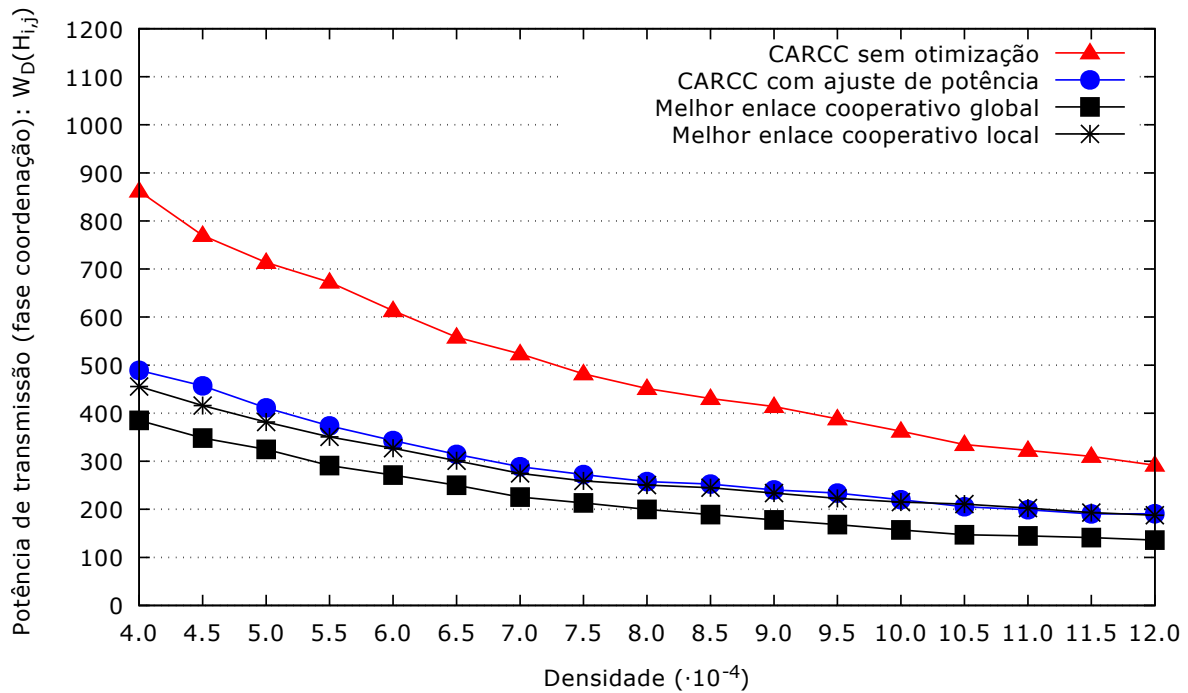


Figura 5.13: CARCC: Potência de transmissão na fase de coordenação.

potência. No melhor caso, a potência requerida para atingir o auxiliar mais distante é de 135,96 unidades para o melhor enlace global, 187,82 unidades para o melhor enlace localizado, 291,82 unidades para o CARCC sem otimização e 190,86 para o CARCC com ajuste de potência. O maior intervalo de confiança é de  $\pm 15,89$ . Como esperado, o valor para o enlace criado com topologia global é menor que os valores produzidos pelos enlaces criados com informação localizada a 2 saltos, pelo CARCC sem otimização e pelo CARCC com ajuste de potência. Este comportamento resulta da possibilidade de encontrar um enlace de menor custo em qualquer região do grafo, enquanto as alternativas localizadas estão restritas à região geográfica próxima a articulação. Para o CARCC sem otimização, a potência de transmissão necessária para atingir o auxiliar mais distante é muito maior que as outras soluções. Isto ocorre devido à distância sobrestimada entre a origem e os componentes de rede desconectados. Neste caso, foi necessário utilizar um maior número de auxiliares, como pode ser notado nos resultados da Métrica  $M3$ , apresentada a seguir. Com um maior número de auxiliares, maior é a potência necessária para os alcançar. Os valores para o CARCC com ajuste de potência e o melhor enlace local são próximos. Este comportamento resulta do fato da origem de seus candidatos a auxiliares serem limitados em número e região geográfica. O CARCC com ajuste de potência requer, aproximadamente, 31,44% mais potência que o melhor enlace global. A diferença entre o melhor enlace criado com informação local e o CARCC com ajuste de potência reduziu de 7,39%, no pior cenário, para 1,62%, no melhor cenário, demonstrando que os resultados



produzidos pelo CARCC se aproximam do melhor enlace local com densidades mais altas. A rotina EDBHSS, usada para selecionar os auxiliares, tem como prioridade os nós mais próximos à origem e utiliza uma estimativa arbitrária muito superior à distância real. A rotina DBHSS adotada pelo CARCC2h como métrica a soma da distância entre a origem e os auxiliares para o destino, ao priorizar os candidatos a auxiliares e, aplicam-se estimativas de distância com maior precisão. Este fato explica a diferença entre os valores obtidos pelo enlace criado pelo CARCC2h apresentado na seção anterior e o melhor enlace local utilizando como referência para o CARCC.

A Figura 5.14 exibe os resultados de simulação para a Métrica  $M2$ . O eixo “y” apresenta a potência média de transmissão individual para a fase de cooperação. São apresentados os mesmos cenários da Figura 5.13 com a mesma configuração para cada linha. No pior caso, a potência individual requerida durante a fase de cooperação é de 2706,52 unidades para o melhor enlace global, 2808,13 unidades para o melhor enlace local, 4000,27 unidades para o CARCC sem otimização e 2841,73 para o CARCC com ajuste de potência. O maior intervalo de confiança é de  $\pm 26,75$ . A média dos valores é 2603,68 unidades para o melhor enlace global, 2697,71 unidades para o melhor enlace local, 3911,90 unidades para o CARCC sem otimização e 2784,22 para o CARCC com ajuste de potência. No melhor caso, a potência requerida para atingir o auxiliar mais distante é de 2544,48 unidades para o melhor enlace global, 2624,62 unidades para o melhor enlace localizado, 3839,71 unidades para o CARCC sem otimização e 2739,67 para o CARCC

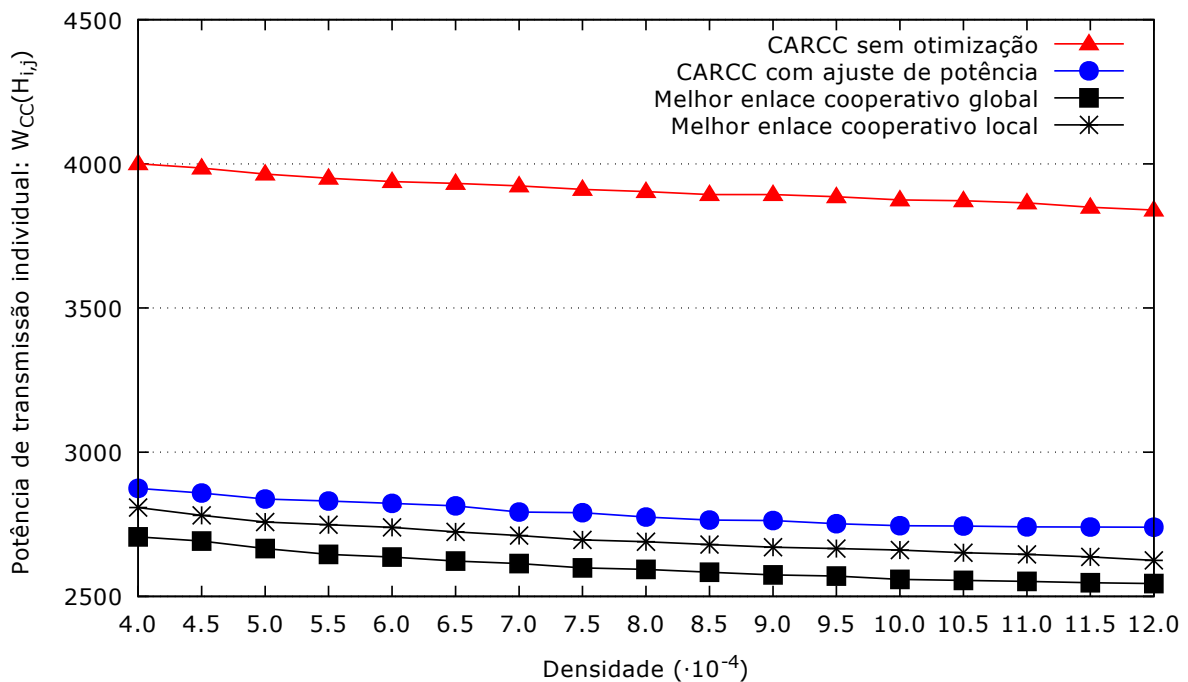


Figura 5.14: CARCC: Potência de transmissão na fase de cooperação.

com ajuste de potência. O maior intervalo de confiança é de  $\pm 25,01$ . Como pode ser observado pela figura, a potência individual de transmissão decresce com o aumento na densidade em um ritmo lento. Com auxiliares mais próximos, maior é a contribuição para a potência recebida no destino. Desta forma, é necessário menos energia individual quando os nós transmitem simultaneamente. Para o CARCC sem otimização, a potência individual de transmissão é muito superior aos outros cenários. Este comportamento resulta da distância sobrestimada entre os componentes de rede e, conseqüentemente, em relação aos nós que participam do enlace cooperativo. O melhor enlace global, o melhor enlace localizado e o CARCC com ajuste de potência apresentam resultados similares. No pior caso, o CARCC requer, no máximo, 7,67% e 4,38% mais potência que o melhor enlace global e local, respectivamente. Além disso, o CARCC necessita que os nós utilizem, no máximo, 56,82% da potência de transmissão máxima individual ( $P_{MAX}$ ) para criar o enlace sobressalente.

A Figura 5.15 representa os resultados de simulação para a Métrica  $M3$ . O eixo “y” corresponde ao número de auxiliares em  $H_{i,j}$ . São apresentados os mesmos cenários da Figura 5.13 com a mesma configuração para cada linha. Exceto pelo CARCC sem otimização, é possível notar que o número de auxiliares  $|H_{i,j}|$  é, aproximadamente, um para todas as densidades. Isto indica que, em geral, é necessária a cooperação apenas da origem e de um auxiliar para criar um enlace cooperativo entre dois componentes ligados por elementos críticos. Para o CARCC sem otimização, é necessário um total de 1,48

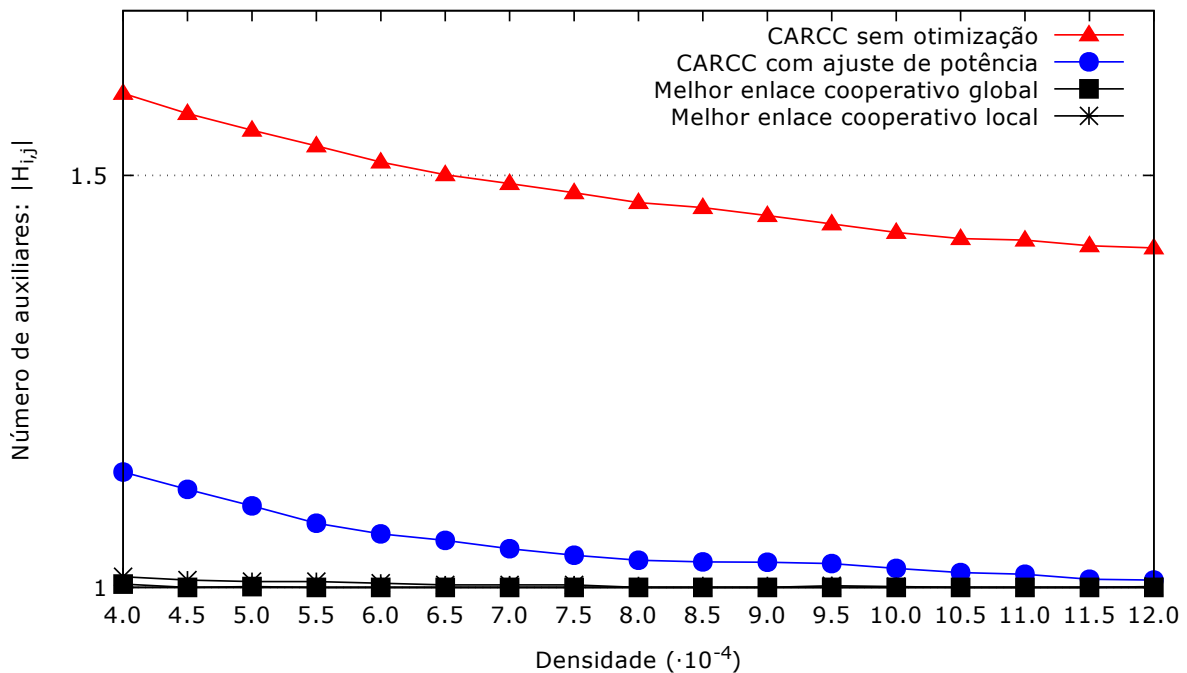


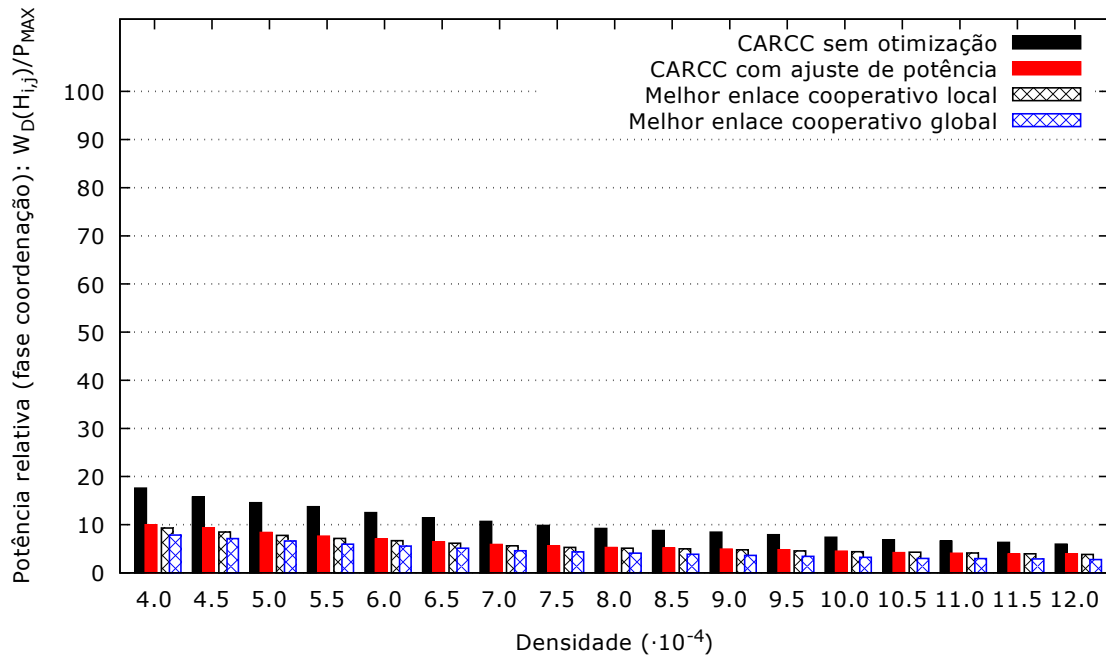
Figura 5.15: CARCC: Número de auxiliares.

auxiliares na média. Para os dados apresentados, o maior intervalo de confiança é de 0,04. Para o CARCC com ajuste de potência o número de auxiliares reduziu de 1,14 para 1,0. Observe que a rotina de seleção de auxiliares, *EDHSS*, interrompe a adição de candidatos ao conjunto de auxiliares assim que determinar que o destino pode receber o sinal adequadamente. Como a origem e seus auxiliares não mudam após a execução das linhas 1 a 9, pode-se inferir que, para os cenários avaliados, a origem selecionada requer apenas um auxiliar para alcançar os componentes biconectados adjacentes, mesmo com a distância sobrestimada.

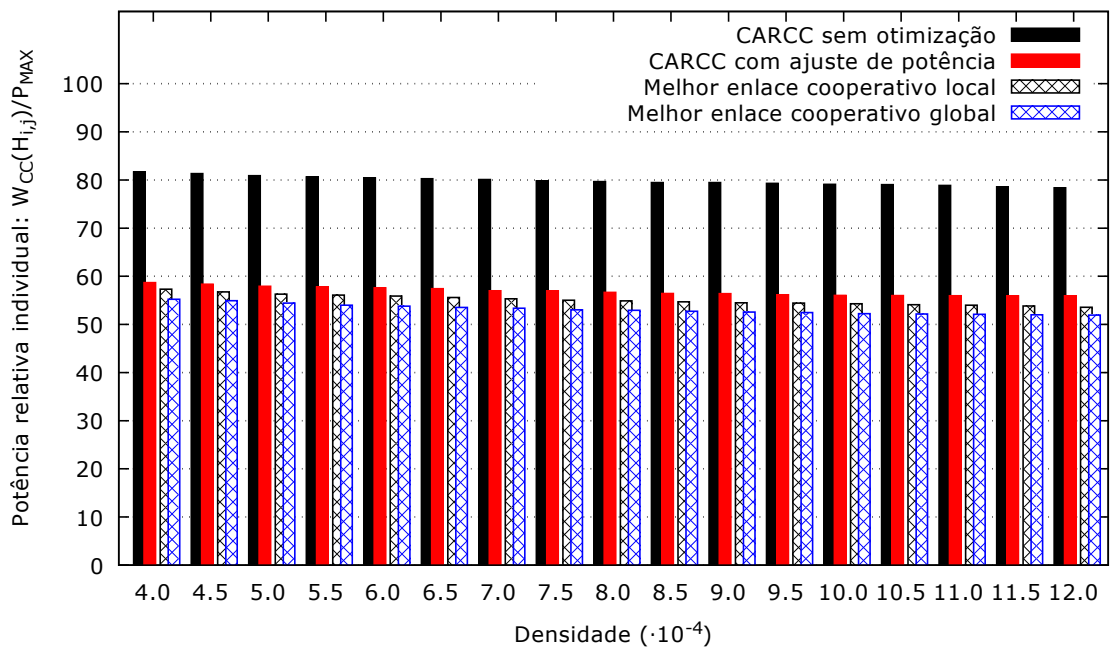
A Figura 5.16 apresenta, no eixo “y”, a relação entre a potência de transmissão utilizada e a potência de transmissão máxima. As barras sólidas representam, a potência utilizada pelo CARCC sem otimização e após o ajuste de potência, respectivamente nas cores vermelha e preta. As barras hachuradas de cor preta e azul representam, respectivamente, a potência necessária para o enlace cooperativo criado por meio de informação global da topologia e com o uso de informação topológica a dois saltos. A Figura 5.16(a) apresenta a porcentagem da potência máxima de transmissão necessária para atingir o nó auxiliar mais distante. A potência de transmissão média é de 10,22%, 5,96%, 4,54% e 5,73%, respectivamente para o CARCC sem otimização, o CARCC com ajuste de potência, o enlace global e para o enlace localizado a 2 saltos. Com o incremento na densidade, a potência de transmissão necessária quando se utiliza o CARCC com ajuste de potência se aproxima dos cenários em que se utiliza informação global da topologia e quando se utiliza informação topológica a 2 saltos. Estes dados também demonstram que, em geral, a origem selecionada pelo CARCC com ajuste de potência conseguem selecionar vizinhos que requerem, no máximo, 1,42% mais energia que os nós selecionados para criar os enlaces cooperativos através de informação global da topologia e com informação topológica a 2 saltos. A Figura 5.16(b) apresenta a porcentagem da potência máxima de transmissão utilizada pelas nós participantes da fase de cooperação. A potência de transmissão média é de 79,83%, 56,82%, 53,14% e 55,06%, respectivamente para o CARCC sem otimização, o CARCC com ajuste de potência, o enlace global e para o enlace localizado a 2 saltos. A diferença média entre o CARCC com ajuste de potência e o enlace localizado a dois saltos é de 3,26% enquanto a maior diferença é de apenas 4,43%. Assim como no CARCC2h, a origem consegue comunicar com o destino, por meio do enlace cooperativo criado com CC em duas fases, utilizando uma quantidade de energia inferior a aquela necessária quando se aplica a potência máxima de transmissão. Os nós auxiliares utilizam, no máximo, 81,64% da potência máxima de transmissão, mesmo sem o ajuste de potência.

A Figura 5.17 apresenta os resultados de simulação para a Métrica *M4*. O eixo “y” representa a média da potência total de transmissão para o enlace cooperativo. Esta métrica corresponde à soma da potência de transmissão para a fase de coordenação (enviar

os dados para os nós auxiliares) para a fase de cooperação (potência utilizadas por todos os nós durante a transmissão dos dados para o destino). São apresentados os mesmos cenários da Figura 5.13 com a mesma configuração para cada linha. No pior caso, a potência total requerida é de 5808,80 unidades para o melhor enlace global, 6108,23 unidades



(a)



(b)

Figura 5.16: CARCC2h: Potência de transmissão relativa.

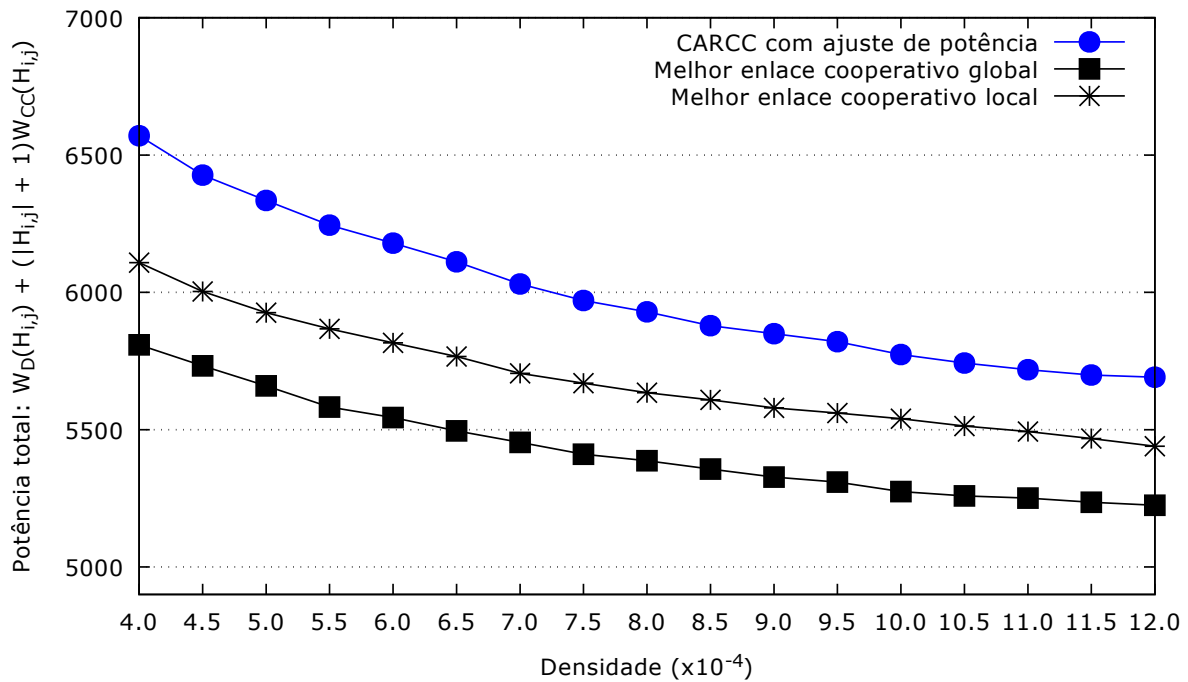


Figura 5.17: CARCC: Potência de transmissão total.

para o melhor enlace local, 11261,13 unidades para o CARCC sem otimização e 6570,42 para o CARCC com ajuste de potência. O maior intervalo de confiança é de  $\pm 248,89$ . A média dos valores é 5430,52 unidades para o melhor enlace global, 5684,96 unidades para o melhor enlace localizado, 10217,79 unidades para o CARCC sem otimização e 5994,91 para o CARCC com ajuste de potência. No melhor caso, a potência requerida para atingir o auxiliar mais distante é de 5224,92 unidades para o melhor enlace global, 5439,68 unidades para o melhor enlace localizado, 9553,45 unidades para o CARCC sem otimização e 5694,87 para o CARCC com ajuste de potência. O maior intervalo de confiança é de  $\pm 167,08$ . A figura não mostra os resultados para o CARCC sem otimização já que sua potência de transmissão total é muito maior que as outras alternativas. Como pode ser visto na figura, a potência de transmissão total reduz com o aumento na densidade. Este comportamento decorre da redução na potência de transmissão necessária para alcançar os auxiliares e na redução da distância entre os componentes de rede. A potência de transmissão necessária pelo CARCC com ajuste de potência é, aproximadamente, 10,35% maior que o melhor enlace global e 5,42% maior que o melhor enlace local. O CARCC com ajuste de potência e o melhor enlace local utilizam, respectivamente, informação a 1 salto e a 2 saltos da topologia da rede. Nos cenários avaliados, o CARCC com otimização reduziu a diferença para o melhor enlace global de 13,11% para 8,99%. Comparado ao melhor enlace local, a diferença reduziu de 7,54% para 4,99%.

A Figura 5.18 apresenta os resultados de simulação para a Métrica  $M5$ . O eixo “y”

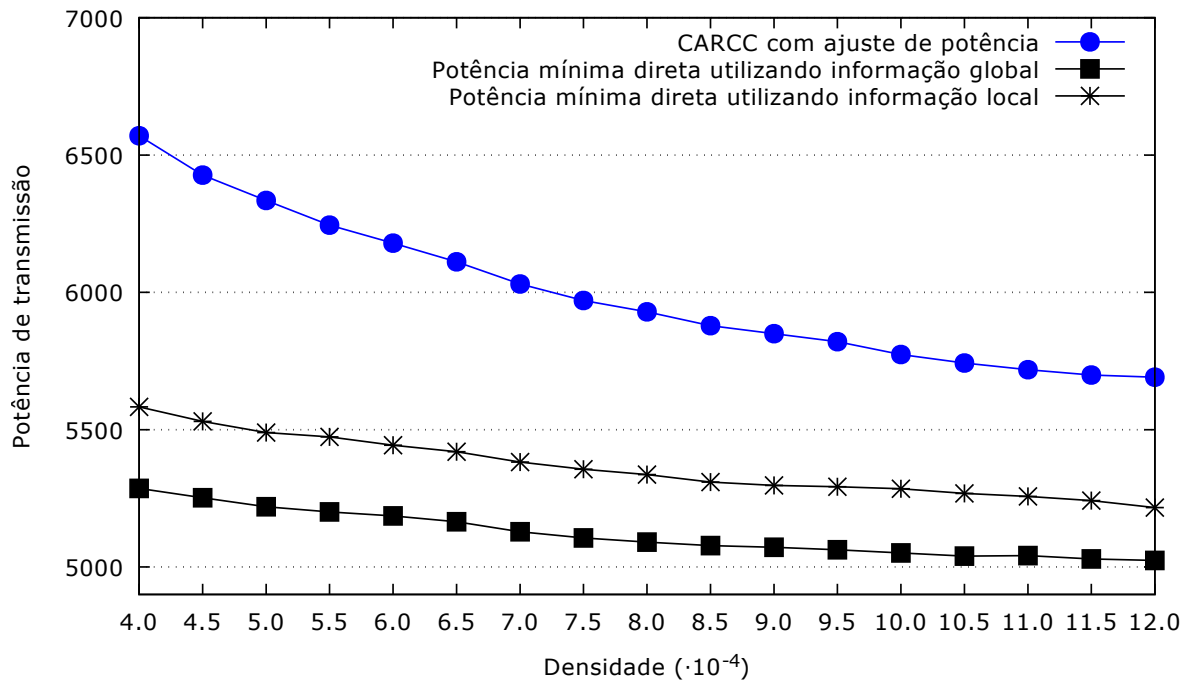


Figura 5.18: CARCC: Potência de transmissão sem o limite  $P_{MAX}$ .

corresponde à potência total de transmissão necessária para restabelecer a conectividade usando enlaces diretos e a potência de transmissão total necessária pelo CARCC com ajuste de potência. Apesar dos nós não poderem ajustar a potência de transmissão além de  $P_{MAX}$  até o limite necessário para restabelecer a conectividade, ele fornece o limiar inferior. A linha marcada com pontos representa a potência de transmissão requerida pelo CARCC com ajuste de potência, enquanto as linhas marcadas com quadrados e estrelas descrevem, respectivamente, a potência necessária para manter a conectividade usando enlaces diretos entre os nós mais próximos, baseado na topologia global e entre os vizinhos das articulações. Assim como esperado a potência de transmissão é superior a 4900 unidades em todos os casos, confirmando que não é possível criar o enlace direto. A potência necessária ao utilizar a topologia local é maior que aquela utilizando informação global, reforçando que o maior número de opções pode levar a melhores resultados. Com aumento na densidade, a potência de transmissão total requerida pelo CARCC com ajuste de potência se aproxima do limite inferior. No pior caso, ela necessita de 24,30% mais potência que o requerido utilizando informação global e 17,63% mais potência quando se usa apenas informação localizada. Com densidade de  $12 \times 10^{-4}$ , a diferença entre o CARCC e a potência de transmissão dos enlaces diretos globais é de 13,36%, ao passo que a diferença é de 9,17% comparado ao cenário em que se utiliza informação local. Note que, com maior potência de transmissão, maior é o consumo de energia. Contudo, manter a conectividade da rede se faz necessário em diversas situações como apresentado

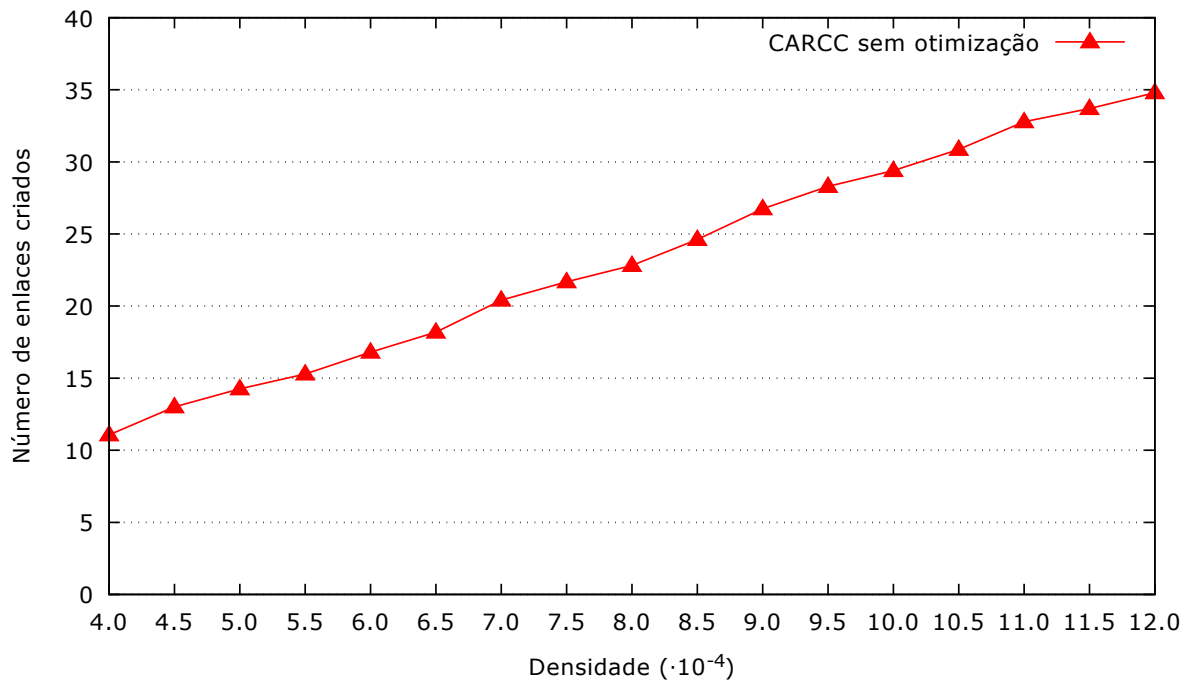


Figura 5.19: CARCC: Número de enlaces criados sem otimização (linhas 1-9).

na introdução deste trabalho.

A Figura 5.19 representa os resultados de simulação para a Métrica  $M6$ . O eixo “y” corresponde ao número de enlaces criados com o CARCC sem otimização. O CARCC com ajuste de potência e os melhores enlaces globais e locais são restritos a um único enlace cooperativo e, portanto, não são apresentados. Esta figura mostra que o CARCC sem ajuste de potência pode criar entre  $11,07 \pm 0,59$  e  $34,79 \pm 2,02$  enlaces cooperativos, na região próxima às articulações, para os parâmetros de simulação utilizados. Este comportamento corrobora que o CARCC pode criar enlaces além dos vizinhos das articulações. Estes enlaces podem ser utilizados para a criação de múltiplos caminhos entre os componentes biconectados. Assim, rotas diferentes podem ser exploradas para permitir que sejam encontradas as de menor custo.

### 5.3 Síntese e Discussão

Este capítulo explorou a Comunicação Cooperativa para oferecer resiliência à conectividade da rede. Foram apresentadas duas soluções baseadas em conhecimento localizado que são capazes de computar enlaces cooperativos entre os componentes de rede que surgem após a falha de articulações. Os algoritmos para localização de elementos críticos e componentes de rede, apresentados no Capítulo 4, foram aplicados para prever a segmen-

tação da rede. Para verificar o desempenho, dados de simulação foram comparados com enlaces criados utilizando informação global da topologia.

A primeira solução, chamada CARCC2h, faz uso de informação localizada a 2 saltos, a partir da articulação, para computar enlaces bidirecionais ou direcionais capazes de recuperar a conectividade da rede. As informações sobre a topologia são coletadas utilizando mensagens especiais ou informações sobre a tabela de roteamento. O algoritmo usa informações sobre a presença de articulações na rede, que pode ser obtida por meio do algoritmo DAS, apresentado no Capítulo 4. O enlace cooperativo é efetivamente criado quando a articulação falha. O monitoramento do estado do elemento crítico é realizado pelos vizinhos selecionados para a criação do enlace cooperativo. Os resultados de simulação indicam que os enlaces cooperativos requerem que os nós utilizem 62,43% da potência máxima de transmissão. Esta solução é capaz de recuperar a conectividade apenas quando a articulação liga dois componentes biconectados da rede.

A segunda solução, chamada CARCC, utiliza uma estimativa de distância arbitrária entre os componentes para determinar se a conectividade pode ser recuperada. O processamento é realizado diretamente pela origem dos enlaces cooperativos (vizinhos das articulações). Como requisito, o CARCC requer que cada nó saiba se é uma articulação e que detenha um identificador para o componente biconectado a que pertence. Estas informações podem ser obtidas ao utilizar, sequencialmente, os algoritmos DAS, DBS e DDBC. A solução usa apenas informação local e, devido ao uso de distâncias arbitrárias sobrestimadas, um grande número de enlaces cooperativos são criados. Então, o algoritmo utiliza um conjunto de mensagens para ajustar a potência de transmissão e manter apenas o enlace de menor custo. Esta solução é capaz de recuperar a conectividade com um aumento de 1,2% na potência individual dos nós em relação ao melhor enlace direcional localizado a 2 saltos (equivalente ao CARCC2h), mesmo em cenários em que uma articulação conecta vários componentes de rede. Ao contrário da solução a 2 saltos, o CARCC disponibiliza o enlace cooperativo antes mesmo da falha no elemento crítico. Assim, além de permitir a recuperação da conectividade, o enlace cooperativo também pode ser utilizado para garantir a biconectividade da rede e a redução na carga de trabalho das articulações.

As soluções de recuperação de conectividade propostas nesta tese utilizam heurísticas simples para a seleção de nós auxiliares. A adição dos candidatos ao conjunto de auxiliares tem como métrica a distância estimada entre os nós. O objetivo é a redução no custo computacional. Neste trabalho, o tempo de vida da rede não foi avaliado. Variações destas heurísticas podem considerar, por exemplo, a energia remanescente dos nós. Durante a coleta de informações sobre a topologia, os nós podem propagar estas informações. Desta forma, os enlaces cooperativos podem ser criados por meio daqueles nós que proporcionam



um maior tempo de vida para a rede.

A redução no custo computacional é a principal contribuição das soluções apresentadas neste capítulo. O requisito de tempo para os algoritmos baseados em Comunicação Cooperativa para garantir a conectividade da rede são quadráticos em relação ao número de vértices [36, 45, 61, 62]. As soluções propostas requerem um tempo quadrático em relação ao grau do grafo. Desta forma, os enlaces cooperativos podem ser criados dinamicamente, durante a operação da rede. Para avaliar as soluções propostas, informações globais da topologia foram utilizadas para criar enlaces cooperativos por meio da busca exaustiva. Os enlaces cooperativos criados pelas propostas apresentam um custo de potência maior. Porém, não foram considerados os custos de energia para coletar dados sobre toda a topologia da rede. Assim, espera-se que o impacto das soluções propostas sejam ainda mais satisfatório. Além disso, as informações topológicas coletadas para obter informação global da topologia devem ser encaminhadas por meio das articulações, reduzindo ainda mais o tempo de vida da rede.

# Capítulo 6

## Conclusão

As aplicação das redes *ad hoc* e de sensores sem fio foram idealizadas como ferramentas para auxiliar missões críticas e temporárias [34, 50, 58], como uma rede de comunicação alternativa em caso de falha nos sistemas públicos [19] e o monitoramento e alerta de desastres naturais [18, 34]. Considerando os aspectos de comunicação a múltiplos saltos, o sucesso da aplicação destas redes, sobretudo em missões críticas, depende da conectividade da rede. Assim, oferecer resiliência à conectividade das redes *ad hoc* e de sensores sem fio podem consolidar estas tecnologias.

O principal objetivo deste trabalho foi apresentar alternativas para a recuperação de conectividade, empregando a Comunicação Cooperativa (CC). Neste sentido, foram apresentados algoritmos para determinar os elementos críticos e os componentes biconectados das redes *ad hoc* sem fio. Esta informação foi utilizada para prever a segmentação da rede, enquanto a CC foi aplicada para estender o alcance dos nós e criar enlaces sobresalentes. A Seção 6.1 apresenta as considerações finais. Em seguida, a Seção 6.2 enumera possibilidades de trabalhos futuros.

### 6.1 Considerações Finais

Este trabalho apresentou uma revisão bibliográfica da localização de elementos críticos em redes *ad hoc* sem fio e como esta informação pode ser aplicada na prevenção da segmentação da rede. Como alternativa para recuperar a conectividade, foram discutidos os aspectos da Comunicação Cooperativa como mecanismos para aumentar o alcance dos nós além do raio máximo de transmissão. Na revisão, foram apresentadas duas soluções que compõem a lista de contribuições deste trabalho. A primeira solução, publicada em [2], aplica a agregação de dados nas articulações ou em seus vizinhos a  $k$ -saltos para aumentar o tempo de vida da rede. Foi possível reduzir em até 28,48% a quantidade de informações trafegadas nas articulações e em até 30,90% o seu consumo de energia. Desta forma, o

tempo de vida da rede e, conseqüentemente, seu particionamento foram postergados. A outra contribuição, publicada em [45], consiste em criar enlaces cooperativos, em Redes de Sensores Sem Fio (RSSF), para permitir a conectividade com o sorvedouro, utilizando enlaces cooperativos que consideram o atraso nos pacotes e o custo de energia. A solução garantiu, para os cenários avaliados, até 85% de conectividade da rede reduzindo em até 60% o número de saltos para o sorvedouro e em 50% o atraso dos pacotes na rede.

Para a localização das articulações, das pontes e dos componentes biconectados, esta tese propõe três algoritmos distribuídos e interdependentes. O primeiro algoritmo, chamado *Distributed Articulation Search* (DAS) ou Busca Distribuída por Articulações, permite a localização das articulações da rede utilizando  $O(|V|)$  mensagens. Apesar de usar uma variação conhecida do algoritmo DDFS, a solução é capaz de encontrar estes elementos empregando a metade do número de mensagens em relação à proposta de Chaudhuri [14]. O segundo algoritmo, chamado *Distributed Bridge Search* (DBS) ou Busca Distribuída por Pontes, emprega as variáveis armazenadas em memória pelo DAS para encontrar as pontes utilizando  $O(|V|)$  mensagens. O terceiro e último algoritmo, chamado *Distributed Define Biconnected Components* (DDBC) ou Busca Distribuída por Componentes Biconectados, faz uso das informações deixadas pelos DAS e DBS para identificar os componentes biconectados da rede. Este algoritmo também utiliza  $O(|V|)$  mensagens. Ao final da execução deste conjunto de algoritmos, cada nó da rede tem conhecimento, por meio de variáveis locais, se é uma articulação, se participa de uma ponte, quais nós que compõem as pontes e uma lista dos componentes biconectados a que pertence. A soma do custo computacional dos três algoritmos equivale à solução apresentada em [14] que é capaz de encontrar apenas as articulações. Os dados de simulação serviram para definir os cenários em que se aplicam a recuperação da conectividade. Os algoritmos DAS e DBS foram publicados no *International Journal of Networking and Computing* [4].

Um vez que os elementos críticos e os componentes biconectados são devidamente identificados, duas propostas podem ser aplicadas para recuperar a conectividade da rede por meio da Comunicação Cooperativa. Na primeira alternativa, chamada *Connectivity Auto-Recovering through Cooperative Communication Using 2-hop Information* (CARCC2h) ou Recuperação Automática de Conectividade Por Meio da Comunicação Cooperativa Utilizando Informações a 2 Saltos, as articulações coletam informações a dois saltos da topologia e computam enlaces cooperativos entre os componentes da rede. Os nós escolhidos, para criar o enlace, são notificados. Quando as articulações falham, o enlace é efetivamente criado. Esta solução utiliza, no máximo,  $O(\Delta(G)^2 \cdot \Lambda)$  unidades de tempo para determinar, a partir das articulações, o melhor enlace cooperativo localizado a dois saltos, onde  $\Delta(G)$  representa o grau da rede e  $\Lambda$  corresponde ao custo para selecionar os nós auxiliares em um enlace cooperativo. Na segunda opção, chamada *Connectivity*

*Auto-Recovering through Cooperative Communication* (CARCC) ou Recuperação Automática de Conectividade por meio da Comunicação Cooperativa Utilizando Informações a 1 Salto, os enlaces cooperativos são computados pelos vizinhos das articulações. Não é necessário coletar informações sobre a topologia da rede, porém as articulações devem enviar uma estimativa arbitrária de distância entre componentes biconectados ligados a ela. Múltiplos enlaces cooperativos são criados, em seguida, um ajuste de potência é realizado para manter apenas o enlace de menor custo. O enlace cooperativo torna-se disponível de imediato tornando possível unir os componentes biconectados em um só e realizar, por exemplo, o balanceamento de carga. Esta solução utiliza, no máximo,  $O(\Delta(G) \cdot \Lambda)$  unidades de tempo para computar os enlaces cooperativos. As soluções apresentadas, para a recuperação de conectividade, se diferenciam daquelas apresentadas na literatura por não necessitarem de informação global sobre a topologia da rede, assim como não requerem a posição geográfica dos nós obtida através de GPS [61, 62].

Dados de simulação demonstram que os enlaces cooperativos criados empregando informação localizada da topologia apresentam um custo maior, em relação a potência de transmissão, do que aqueles enlaces criados ao considerar a topologia global. Apesar de apresentarem custo maior, demonstrou-se que a cooperação requer que os nós utilizem, no máximo, a 63% da potência máxima para manter os enlaces localizados. É importante notar que, apesar do maior custo dos enlaces localizados, é necessário garantir a conectividade da rede para os cenários idealizados. Os resultados obtidos indicam que a Comunicação Cooperativa é uma alternativa importante para oferecer resiliência à conectividade das rede *ad hoc* sem fio. Parte dos resultados obtidos com o CARCC2h foram publicados em artigos de conferência [3, 6, 7]. Uma versão final foi publicada na Revista IEEE América Latina [5].

## 6.2 Trabalhos Futuros

Considerado a revisão no estado da arte e as soluções propostas por este trabalho, foram concebidas as seguintes sugestões para trabalhos futuros:

1. **Integração da recuperação de conectividade com algoritmo de roteamento:** As soluções propostas neste trabalho dependem das articulações para prever a segmentação da rede. Por outro lado, situações adversas como deslizamentos e enchentes podem remover completamente os dispositivos em uma determinada região geográfica. Este cenário pode produzir diversos componentes de rede e as articulações não podem ser utilizadas para prever a segmentação. Da mesma forma, as soluções existentes na literatura, como dispor de novos nós ou movê-los não será

eficaz, já que, em geral, não será possível coletar informações sobre a nova topologia. Uma possível solução para este problema é detectar a impossibilidade de encaminhar pacotes para o sorvedouro, por meio de informações dos algoritmos de roteamento. Os nós podem definir um identificador para aquele componente isolado e se organizarem para criar enlaces cooperativos que alcancem nós com identificadores diferentes. A técnica pode ser incorporada como mecanismo de recuperação de rota. Como ponto de partida, Pereira e Moraes [40] apresentam uma lista de mecanismos para recuperar a rota no *Ad Hoc On-Demand Distance Vector* (AODV).

2. **Garantir redes biconectadas:** Esta tese explorou o uso da CC para a recuperação de conectividade a partir das articulações. Em especial, o CARCC permite a criação de múltiplos enlaces entre os componentes biconectados. Conforme discutido no Capítulo 2, um componente biconectado não possui articulações. Assim, pequenas alterações podem ser realizadas na proposta para tentar obter uma rede biconectada por meio de enlaces cooperativos criados a partir dos vizinhos das articulações. Em especial, podem ser analisados os benefícios referentes ao balanceamento de carga e tempo de vida da rede.
3. **Efeito de borda sobre a presença de elementos críticos:** Os resultados apresentados no Capítulo 4 indicam que o número de nós e enlaces críticos da rede é diferente quando se considera o efeito de borda. Trabalhos relacionados já demonstraram que o grau dos nós não é homogêneo, sobretudo na região de borda da rede [31, 43]. Assim, sugere-se o estudo empírico da região de borda sobre o número de elementos críticos e a elaboração de modelos probabilísticos semelhantes ao proposto por Yong e Hu [59] que consideram o efeito de borda.
4. **Controle de Topologia orientado a biconectividade:** Ramanathan e Rosales-Hain [44] introduziram o uso de uma técnica similar ao MST para realizar o CT. Contudo, os autores também discutiram a importância de se obter a biconectividade da rede e apresentaram uma segunda etapa que incrementa o número de arestas até que existam ciclos entre os nós. Este trabalho pode ser estendido em diversos aspectos. Primeiro, o trabalho não considera, ao incrementar a conectividade, outras formas de controle de topologia. É possível, portanto, explorar as características das topologias resultantes, das diferentes técnicas de controle, e demonstrar que elas tem maior e menor capacidade de resiliência e são suscetíveis a um esforço diferente e se tornarem biconectadas. Em segundo lugar, o trabalho não prevê a forma e o esforço para localizar as articulações sobretudo de forma recorrente. Pode-se averiguar, por exemplo, que técnicas que produzem inúmeras articulações (como aquelas baseadas

em árvores) podem oferecer as informações necessárias para guiar a inclusão de arestas para se obter uma topologia biconectada.

Além dos trabalhos sugeridos, as propostas apresentadas neste trabalho também podem ser exploradas sobre outros aspectos. Os enlaces cooperativos podem ser criados considerando métricas, como por exemplo, a energia remanescente dos nós. Quando múltiplos enlaces cooperativos são criados, antes ou depois da falha nos elementos críticos, a seleção dos enlaces que permanecem ativos podem ser utilizados para favorecer os caminhos. Neste sentido, as métricas *Power Stretch Factor* e *Distance Stretch Factor* [62] podem ser aplicadas para comparar as topologias geradas ou os trabalhos relacionados.

# Referências

- [1] Salah Abdulhadi, Muhammad Jaseemuddin, e Alagan Anpalagan. A Survey of Distributed Relay Selection Schemes in Cooperative Wireless Ad hoc Networks. *Wireless Personal Communications*, 63(4):917–935, 2012. 33
- [2] Ulisses R. Afonseca, Paulo H. Azevêdo Filho, Jacir L. Bordim, e Priscila S. Barreto. Localização e Redução do Consumo de Energia em Pontos de Articulação em Redes de Sensores Sem Fio. In *II Workshop de Sistemas Distribuídos Autônomicos*, pages 21–24, 2012. 2, 22, 24, 25, 26, 105
- [3] Ulisses R. Afonseca, Thiago F. Neves, e Jacir L. Bordim. Connectivity recovering using cooperative links in wireless networks. In *5th International Workshop on Parallel and Distributed Algorithms and Applications. The First International Symposium on Computing and Networking - Across Practical Development and Theoretical Research - Dogo SPA Resort, Matsuyama, Japan, Dec 2013*. 107
- [4] Ulisses R. Afonseca, Thiago F. Neves, e Jacir L. Bordim. Employing cooperative communication to recover network connectivity in ad hoc networks. *International Journal of Networking and Computing*, 4:336, 2014. 106
- [5] Ulisses R. Afonseca, Thiago F. Neves, e Jacir L. Bordim. Melhoria da conectividade de rede sem fio por meio de comunicação cooperativa. *Revista IEEE América Latina*, 12:746, 2014. 107
- [6] Ulisses R. Afonseca, Thiago F. Neves, e Jacir L. Bordim. Sinkpath: A cooperative communication protocol to reestablish wsn connectivity. In *Proceedings of 13th International Conference on WWW/Internet 2014*. IADIS Press, Oct 2014. 22, 107
- [7] Ulisses R. Afonseca, Thiago F. Neves, e Jacir L. Bordim. Wireless ad hoc networks resilience through cooperative communication. In *Proceedings of ICSNC 2014, The Ninth International Conference on Systems and Networks Communications. Nice, France, Oct 2014*. 22, 107
- [8] Joan M. Aldous e Robin J. Wilson. *Graphs and Applications: An Introductory Approach*. Undergraduate Mathematics Series. Springer Science & Business Media, 2003. 9, 10
- [9] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing - MobiHoc '02*, page 80, 2002. 1, 65

- [10] Yogesh Bhute e Ar Raut. A Survey on Relay Selection Strategies in Cooperative Wireless Network for Capacity Enhancement. *International Journal of Computer Applications*, 65(25):12–17, 2013. 33
- [11] Mihaela Cardei, Jie Wu, e Shuhui Yang. Topology control in ad hoc wireless networks using cooperative communication. *IEEE Mobile Computing*, 5(6):1–14, 2006. 2, 31, 34, 38, 44
- [12] Mihaela Cardei, Jie Wu, e Shuhui Yang. Topology control in ad hoc wireless networks using cooperative communication. *IEEE Transactions on Mobile Computing*, 5(6):711–724, 2006. 39
- [13] Pranay Chaudhuri. An optimal distributed algorithm for computing bridge-connected components. *The Computer Journal*, 40(4), 1997. 1, 19, 61, 67
- [14] Pranay Chaudhuri. An optimal distributed algorithm for finding articulation points in a network. *Computer Communications*, 21(18):1707–1715, 1998. xii, 1, 3, 4, 6, 10, 13, 16, 19, 20, 21, 22, 24, 48, 52, 61, 67, 72, 106
- [15] Yan Chen, Guanding Yu, Peiliang Qiu, e Zhaoyang Zhang. Power-Aware Cooperative Relay Selection Strategies in Wireless Ad Hoc Networks. *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2006. 34
- [16] Brent N. Clark e Charles J. Colbourn. Unit disk graphs.pdf. *Discrete Mathematics*, 86:165–177, 1990. 15
- [17] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, e Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001. 9, 10, 11
- [18] Governo Federal do Brasil. Ministério da Ciência e Tecnologia. CEMADEM - Centro Nacional de Monitoramento de Alertas de Desastres Naturais, 2013. <http://www.cemaden.gov.br/>. Acessado em 20 de setembro de 2013. 1, 105
- [19] Niloy Ganguly. Post-disaster situation analysis and resource management using delay-tolerant peer-to-peer wireless networks, 2013. [http://cse.iitkgp.ac.in/resgrp/cnerg/disaster\\_dtn/](http://cse.iitkgp.ac.in/resgrp/cnerg/disaster_dtn/). Acessado em 20 de setembro de 2013. 1, 105
- [20] Devendra Goyal e James Caffery Jr. Partitioning avoidance in mobile ad hoc networks using network survivability concepts. In *Proceedings of Seventh International Symposium on Computers and Communications - ISCC'2002.*, pages 553–558. IEEE, 2002. 1, 2, 4, 16, 17, 22, 23, 26
- [21] Y-W Peter Hong, Wan-Jen Huang, e C-C Jay Kuo. *Cooperative Communications and Networking: Technologies and System Design*. Springer Science+Business Media, 2010. 2, 15, 30, 32, 34, 35



- [22] John Hopcroft e Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973. xii, 1, 12, 13, 16, 17, 19, 48, 52, 57, 66
- [23] Tauseef Jamal e Paulo Mendes. Relay selection approaches for wireless cooperative networks. *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob'2010*, pages 661–668, 2010. 34
- [24] Milenko Jorgic, Michaël Hauspie, David Simplot-Ryl, Ivan Stojmenovic, et al. Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks. In *Proceedings of the 3rd IFIP Mediterranean Ad Hoc Networking Workshop*, pages 360–371, Bodrum, Turkey, June 2004. 1, 4, 16, 17, 18, 21, 22, 72
- [25] Benahmed Khelifa, Hafid Haffaf, Merabti Madjid, e David Llewellyn-Jones. Monitoring connectivity in wireless sensor networks. In *IEEE Symposium on Computers and Communications, 2009. ISCC'2009*, pages 507–512, 2009. xii, 1, 2, 4, 16, 17, 22, 23, 24, 26
- [26] Ines Khoufi, Pascale Minet, Anis Laouti, Saoucene Mahfoudh, Ines Khoufi, Pascale Minet, Anis Laouti, Saoucene Mahfoudh, Ines Khoufi, e Pascale Minet. Survey of Deployment Algorithms in Wireless Sensor Networks: Coverage and Connectivity Issues and Challenges. *International Journal of Autonomous and Adaptive Communications Systems (IJAACS)*, 24, 2014. 15
- [27] Mikko Kohvakka, Jukka Suhonen, Marko Hännikäinen, e Timo D Hämäläinen. Transmission power based path loss metering for wireless sensor networks. In *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pages 1–5, Sept 2006. 72
- [28] Henri Koskinen. Statistical model describing connectivity in ad hoc networks. *Wi-Opt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 3–4, 2003. 82
- [29] J. Nicholas Laneman. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Information Theory*, 50(12):3062–3080, December 2004. 2, 31
- [30] J. Nicholas Laneman e Gregory W. Wornell. An efficient protocol for realizing distributed spatial diversity in wireless ad-hoc networks. *Proc. of ARL FedLab Symposium on Advanced Telecommunications and Information Distribution*, pages 294–, 2001. 31, 34
- [31] Luiz Augusto Laranjeira e Genáina Nunes Rodrigues. Border effect analysis for reliability assurance and continuous connectivity of wireless sensor networks in the presence of sensor failures. *IEEE Transactions on Wireless Communications*, 13(8):4232–4246, Aug 2014. 1, 27, 65, 108
- [32] Ning Li e Jennifer C. Hou. Improving connectivity of wireless ad hoc networks. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005*, pages 314–324, July 2005. 2

- [33] Ning Li, Jennifer C. Hou, e Lui Sha. Design and analysis of an MST-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, May 2003. 38
- [34] Anthony Lo, Liang Xia, e Ignas Niemegeers. Europcom-An ultra-wideband (UWB)-based ad hoc network for emergency applications. *Vehicular Technology Conference*, pages 6–10, 2008. 1, 105
- [35] Russell Merris. *Graph Theory*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011. 9, 10, 11
- [36] Thiago Fernandes Neves e Jacir Luiz Bordim. Topology control in cooperative ad hoc wireless networks. *Electronic Notes in Theoretical Computer Science*, 302(0):29 – 51, 2014. Proceedings of the XXXIX Latin American Computing Conference (CLEI 2013). xiii, 2, 6, 37, 42, 43, 44, 67, 75, 104
- [37] Jian Ni e Stephen Anthony Gerard Chandler. Connectivity properties of a random radio network. *IEE Proceedings in Communications*, 141(4):289–296, 1994. 1, 65, 66, 82
- [38] Ahmadreza Nosratinia, Aria; Hunter, Todd E. ; Ahmadreza. Cooperative communication in wireless networks. *Communications Magazine*, 42(October):74–80, 2004. 29, 30, 31, 34
- [39] Oystein Ore. *Graphs and Their Uses*. The Mathematical Association of America, 1990. 9, 10
- [40] Nadilma C. V. N. Pereira e Renato M. de Moraes. Comparative Analysis of AODV Route Recovery Mechanisms in Wireless Ad Hoc Networks. *IEEE Latin America Transactions*, 8(4):385–393, 2010. 108
- [41] Charles E. Perkins e Elizabeth M. Belding-Royer. Ad-hoc on-demand distance vector routing. In *Proceedings. WMCSA '99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999. 45, 72
- [42] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, e Mahesh K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications*, 8(1):16–28, Feb 2001. 72
- [43] Thomas K. Philips, Shiivendra S. Panwar, e Asser N. Tantawi. Connectivity properties of a packet radio network model. *IEEE Transaction on Information Theory*, 35:1044–1047, 1989. 1, 27, 65, 108
- [44] Ram Ramanathan e Regina Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2:404–413, 2000. 16, 38, 108

- [45] Camila F. Rego, Thiago F. Neves, Ulisses R. Afonseca, e Jacir L. Bordim. Exploring cooperative communication to establish energy-and-delay aware routes in wsns. In *Second International Symposium on Computing and Networking (CANDAR), 2014.*, pages 205–211, Dec 2014. 6, 22, 42, 44, 104, 106
- [46] Edward M. Reingold, Jurg Nievergelt, e Narsingh Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice Hall College Div, 1977. 8, 10, 15
- [47] Edward M. Reingold, Jurg Nievergelt, e Narsingh Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice Hall College Div, 1977. 10
- [48] Witte Robert S. e John S. Witte. *Estatística*. LTC - Livros Técnicos e Científicos Editora S.A., 2005. 75
- [49] Ahmed K Sadek, Weifeng Su, e KJ Ray Liu. Multinode cooperative communications in wireless networks. *Signal Processing, IEEE Transactions on*, 55(1):341–355, 2007. 31
- [50] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons Ltd, 2005. 1, 16, 38, 105
- [51] Robert Sedgewick e Kevin Wayne. *Algorithms*. Pearson Education, fourth edition, 2011. 9, 10, 12
- [52] Sanquan Song, Dennis L. Goeckel, e Don Towsley. Collaboration improves the connectivity of wireless networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings.*, 2006. 2
- [53] Ivan Stojmenovic, David Simplot-Ryl, e Amiya Nayak. Toward scalable cut vertex and link detection with applications in wireless ad hoc networks. *IEEE Network*, 25(1):44–48, 2011. 1, 8, 17, 24
- [54] Robert Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2):146–160, June 1972. xii, 1, 10, 12, 13, 14, 16, 21
- [55] David. Tse e Pramod. Viswanath. *Fundamentals of Wireless Communication*. Wiley series in telecommunications. Cambridge University Press, 2005. 29, 30
- [56] Volker Turau. Computing bridges, articulations, and 2-connected components in wireless sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 164–175. Springer, 2006. 4, 15, 16, 21, 22, 24, 52, 87
- [57] Kamin Whitehouse, Chris Karlof, e David Culler. A practical evaluation of radio signal strength for ranging-based localization. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(1):41–52, January 2007. 72
- [58] Jennifer Yick, Biswanath Mukherjee, e Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008. 1, 15, 105

- [59] Tao Yong e Gong Zheng Hu. The articulation nodes modeling of wireless sensor networks. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 12, pages V12–75. IEEE, 2010. 1, 2, 8, 15, 24, 38, 108
- [60] Jieun Yu, Heejun Roh, e Wonjun Lee. Topology Control in Cooperative Wireless. *IEEE Journal on Selected Areas in Communications*, 30(9):1771–1779, 2012. 2, 6, 35, 39, 67
- [61] Jieun Yu, Heejun Roh, Wonjun Lee, Sangheon Pack, e Ding-Zhu Du. Cooperative bridges: Topology control in cooperative wireless ad hoc networks. In *Proceedings of IEEE INFOCOM 2010*, pages 1–9. IEEE, 2010. xiii, 2, 6, 14, 29, 37, 38, 39, 40, 41, 42, 43, 44, 61, 67, 75, 76, 104, 107
- [62] Ying Zhu, Minsu Huang, Siyuan Chen, e Yu Wang. Energy-efficient topology control in cooperative ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(8):1480–1491, 2012. 2, 6, 29, 33, 34, 35, 36, 37, 38, 41, 44, 61, 67, 75, 76, 104, 107, 109