



**Universidade Estadual de Campinas
Instituto de Computação**



Helder May Nunes da Silva Oliveira

**Proteção em Redes Ópticas Elásticas com Multiplexação
Espacial**

CAMPINAS
2018

Helder May Nunes da Silva Oliveira

Proteção em Redes Ópticas Elásticas com Multiplexação Espacial

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Nelson Luis Saldanha da Fonseca

Este exemplar corresponde à versão final da Tese defendida por Helder May Nunes da Silva Oliveira e orientada pelo Prof. Dr. Nelson Luis Saldanha da Fonseca.

CAMPINAS
2018

Agência(s) de fomento e nº(s) de processo(s): CAPES; CNPq, 165446/2015-3

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Márcia Pillon D'Aloia - CRB 8/5180

OL4p Oliveira, Helder May Nunes da Silva, 1989-
Proteção em redes ópticas elásticas com multiplexação espacial / Helder May Nunes da Silva Oliveira. – Campinas, SP : [s.n.], 2018.

Orientador: Nelson Luis Saldanha da Fonseca.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Redes ópticas elásticas. 2. Redes ópticas. 3. Agregação de tráfego. 4. Roteamento multicaminho. I. Fonseca, Nelson Luis Saldanha, 1961-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Protection in space division multiplexing elastic optical networks

Palavras-chave em inglês:

Elastic optical networks

Optical networks

Traffic grooming

Multi path routing

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Nelson Luis Saldanha da Fonseca [Orientador]

Gustavo Bittencourt Figueiredo

Regina Melo Silveira

Edmundo Roberto Mauro Madeira

Fábio Luiz Usberti

Data de defesa: 12-09-2018

Programa de Pós-Graduação: Ciência da Computação



Universidade Estadual de Campinas
Instituto de Computação



Helder May Nunes da Silva Oliveira

Proteção em Redes Ópticas Elásticas com Multiplexação Espacial

Banca Examinadora:

- Prof. Dr. Nelson Luis Saldanha da Fonseca
Instituto de Computação - UNICAMP
- Prof. Dr. Edmundo Roberto Mauro Madeira
Instituto de Computação - UNICAMP
- Dr. Fábio Luiz Usberti
Instituto de Computação - UNICAMP
- Prof. Dr. Gustavo Bittencourt Figueiredo
Instituto de Matemática - Universidade Federal da Bahia
- Profa. Dra. Regina Melo Silveira
Escola Politécnica da Universidade de São Paulo

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 12 de setembro de 2018

Dedicatória

Dedico este trabalho à minha querida mãe Maria e ao meu pai Heli.

Às minhas irmãs Helen, Helece e Helissa.

Ao meu irmão Helton.

E à minha esposa Lais.

*Quem pensa ver algo sem falhas
pensa naquilo que nunca existiu,
que não existe,
e que nunca existirá.*
(Alexander Pope)

Agradecimentos

Agradeço imensamente aos meus pais por terem sempre me encorajado com os estudos e me proporcionado o apoio necessário para chegar até aqui.

Aos meus queridos irmãos que mesmo distantes nunca deixaram de estar presentes. Obrigado por ouvirem minhas queixas e frustrações e por acreditarem em mim.

Agradeço especialmente a minha esposa por todo o seu apoio e inspiração, por sua espera, paciência, compreensão e companheirismo nessa longa jornada.

Ao meu orientador pela orientação, apoio e incentivo oferecidos. Este trabalho não teria sido possível sem o constante apoio fornecido por ele.

Meu muito obrigado a todos os colegas do Laboratório de Redes de Computadores (LRC), tanto aqueles que já não fazem mais parte do LRC quanto aos que ficaram concluindo suas jornadas, sem vocês seria impossível chegar até aqui. As discussões técnicas com todos sempre foram muito frutíferas e sempre trouxeram boas ideias.

Sou grato à família Campineira, família formada diante das dificuldades, mas também nas comemorações e vitórias do dia a dia. Em especial à pequena Helena que com seu sorriso alegrou meus fins de semana.

Agradeço também ao CNPq, Fapesp e a Capes pelo apoio financeiro.

Gostaria também de reconhecer a contribuição de todos aqueles que estão diretamente ou indiretamente associados comigo. Expresso meus sinceros agradecimentos a todos.

Finalmente, um agradecimento muito especial Àquele que mais me ajudou, me inspirou, me acolheu em todos os momentos e que nunca faltou quando Dele eu mais precisava. Agradeço ao Espírito Santo de Deus que Sempre seja feita a Sua vontade.

Resumo

A multiplexação por divisão espacial é uma solução promissora para que as redes ópticas elásticas possam lidar com o esgotamento esperado da capacidade das redes de único núcleo. A introdução da multiplexação por divisão espacial em redes ópticas traz novos desafios para proteção de redes, uma vez que um caminho de luz pode abranger uma alta capacidade e transmitir dados a diferentes taxas. Adicionalmente, a enorme quantidade de tráfego nestas redes provocam a necessidade de proteção contra falhas, uma vez que essas são relativamente frequentes; atualmente, a taxa de falha de um corte de fibra é uma a cada quatro dias. Muitos estudos sobre redes ópticas têm sido desenvolvidos e relatados na literatura. No entanto, apenas recentemente, o estudo de multiplexação espacial para redes ópticas elásticas tem sido considerado. Nesse contexto, embora algoritmos de roteamento e alocação de núcleo e espectro tenham sido propostos na literatura, poucos trabalhos consideram proteção. Além disso, o compartilhamento de recursos de backup não é considerado.

Nesta tese, propõe-se soluções de proteção em redes ópticas elásticas com multiplexação espacial, visando a redução do bloqueio de requisições para estabelecimento de conexão, o melhoramento da utilização dos recursos em redes ópticas elásticas com multiplexação espacial, considerando diferentes cenários de carga e topologias. Para tal, o problema de proteção destas redes levará em consideração a utilização de caminhos de proteção, diferentes formatos de modulação, o uso de agregação de tráfego, o uso de sobreposição de espectro em caminhos de proteção, interferência mínima e roteamento multicaminho. Diversos algoritmos foram propostos e avaliados para prover 100% de proteção contra ocorrência de uma falha, bem como um algoritmo para proteção contra duas falhas simultâneas. Os resultados indicam que os algoritmos propostos produzem um melhor desempenho quando comparado ao desempenho dos algoritmos existentes na literatura.

Abstract

Spatial division multiplexing is a promising solution proposed for elastic optical networks to cope with the expected depletion of the capacity of single core networks. The introduction of space division multiplexing in optical networks brings new challenges for network protection since a lightpath can span high capacity and transmit data at different rates. In addition, there is a great need for protection mechanisms against failure due to the high volume of traffic carried in these networks. Failure is quite frequent in operational optical networks, it is estimated that there is a failure every four hours. Several studies on optical networks have been carried out but, only recently, the study of spatial division multiplexing for elastic optical networks has been considered. In this context, although routing, spectrum, and core allocation algorithms have been proposed in the literature, only a few papers consider protection. In addition, sharing of backup resources is not considered.

In this thesis, it is proposed to provide protection solutions in spatial division multiplexing elastic optical networks, for reducing the blocking of requests for connection establishment, as well as improving the use of resources, considering different load scenarios and topologies. For this, the protection problem of these networks will take into consideration the use of protection paths, different modulation formats, the use of traffic grooming, the use of spectrum overlap in protection paths, minimum interference and multipath routing. Several algorithms are proposed in this thesis to provide 100% protection against a single failure and one algorithm for two simultaneous failure. results indicate that these algorithms overperform existing ones.

Lista de Figuras

1.1	Visão geral das características dos algoritmos desenvolvidos	25
2.1	Previsão do crescimento de tráfego na Internet	29
2.2	Três possibilidades diferentes para provisionamento de recursos.	31
2.3	Esquemas de representação para redes ópticas SDM	32
2.4	Exemplo de proteção de enlace.	36
2.5	Exemplos de proteção de caminho.	37
2.6	Exemplo de proteção de segmento.	38
2.7	Exemplo de enlace transzonal e no ciclo.	39
2.8	Exemplo de <i>p-cycle</i> FIPP.	40
2.9	Exemplo de agregação de tráfego em EONs.	42
2.10	Exemplo de sobreposição de espectro EONs.	42
2.11	Exemplo de multigrafo.	43
3.1	Transformando multigrafo em grafos.	48
3.2	Topologias.	56
3.3	Bloqueio de banda dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.	58
3.4	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.	59
3.5	Razão de fragmentação dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.	59
3.6	Eficiência Energética dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.	60
3.7	Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.	61
3.8	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.	61
3.9	Bloqueio de banda dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.	62
3.10	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.	63
3.11	Razão de fragmentação dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.	64
3.12	Eficiência Energética dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.	65

3.13	Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.	65
3.14	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.	66
3.15	Bloqueio de banda dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.	67
3.16	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.	68
3.17	Razão de fragmentação dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.	68
3.18	Eficiência Energética dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.	69
3.19	Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.	70
3.20	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.	70
4.1	Eficiência x Alcance de transmissão.	72
4.2	Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.	82
4.3	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.	83
4.4	Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.	84
4.5	Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.	85
4.6	Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.	85
4.7	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.	86
4.8	Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.	87
4.9	<i>crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.	88
4.10	Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.	89
4.11	Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.	89

4.12	Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.	90
4.13	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.	91
4.14	Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.	92
4.15	<i>crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.	92
4.16	Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.	93
4.17	Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, PERFECTA, BARTRMAN e MMIFMCC em função da carga da rede para topologia NSF.	94
4.18	Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.	95
4.19	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.	96
5.1	Bloqueio de banda dos algoritmos Cap-DPP, SSCA, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.	107
5.2	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.	108
5.3	Razão de fragmentação dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.	109
5.4	Eficiência Energética dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.	110
5.5	Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.	110
5.6	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.	111
5.7	Bloqueio de banda dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.	112
5.8	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.	113
5.9	Razão de fragmentação dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.	114

5.10	Eficiência Energética dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.	114
5.11	Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.	115
5.12	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.	116
5.13	Bloqueio de banda dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.	117
5.14	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.	118
5.15	Razão de fragmentação dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.	118
5.16	Eficiência Energética dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.	119
5.17	Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.	120
5.18	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.	121
6.1	Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.	131
6.2	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.	132
6.3	Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.	133
6.4	Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.	134
6.5	Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.	134
6.6	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.	135
6.7	Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.	136

6.8	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.	137
6.9	Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.	138
6.10	Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.	139
6.11	Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.	139
6.12	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.	140
6.13	Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.	141
6.14	<i>Crosstalk</i> por <i>slot</i> dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.	142
6.15	Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.	143
6.16	Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.	144
6.17	Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.	144
6.18	Número médio de saltos do caminho de <i>backup</i> dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.	145
7.1	Bloqueio de banda dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.	156
7.2	<i>Crosstalk</i> por <i>slot</i> dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.	157
7.3	Razão de fragmentação dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.	158
7.4	Eficiência Energética dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.	159

7.5	Número médio de saltos dos caminhos primários por requisição dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.	160
7.6	Número médio de saltos dos caminhos de <i>backup</i> dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, Pan-European.	161
7.7	Bloqueio de banda dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.	161
7.8	<i>Crosstalk</i> por <i>slot</i> dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.	162
7.9	Razão de fragmentação dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.	163
7.10	Eficiência Energética dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.	164
7.11	Número médio de saltos do caminho primário por conexão dos algoritmos MOFIO, MINION e PRIME em função da carga da rede para topologia USA.	164
7.12	Número médio de saltos do caminho de <i>backup</i> por conexão dos algoritmos MOFIO, MINION e PRIME em função da carga da rede para topologia USA.	165
7.13	Bloqueio de banda dos algoritmos MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.	166
7.14	<i>Crosstalk</i> por <i>slot</i> dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.	167
7.15	Razão de fragmentação dos algoritmos dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.	167
7.16	Eficiência Energética dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.	168
7.17	Número médio de saltos do caminho primário dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.	169
7.18	Número médio de saltos do caminho de <i>backup</i> por requisição dos algoritmos dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.	169

Lista de Tabelas

3.1	Comparação dos trabalhos relacionados com os algoritmos FIPPMC, SBPPMC, MIFMC.	47
4.1	Características de Modulação	73
4.2	Comparação dos trabalhos relacionados com os algoritmos BARTRMAN, PERFECTA, MMIFMC.	75
5.1	Comparação dos trabalhos relacionados com os algoritmos SBRC, STOP, FRSCA e SSSPF.	99
6.1	Comparação dos trabalhos relacionados com os algoritmos RAMBO, APPROVE e MAROTO.	123
7.1	Comparação dos trabalhos relacionados com os algoritmos MOFIO, MINION e PRIME.	149

Lista de Abreviações e Siglas

AM Adaptive Modulation

APS Automatic Protection Switching

BBR Bandwidth Blocking Ratio

BPSK Binary Phase Shift Keying

CCFs Coupled-core fibers

CCL Connected Component Labelling

CpS Crosstalk per Slot

DPP Dedicated Path Protection

EE Energy Efficiency

EON Elastic Optical Network

ESPAC Elastic Separate-Protection-At-Connection

FGB Filter Guard Band

FIPP Failure-Independent Path-Protecting

FIT Failures In Time

FMF Few-Mode Fiber

FR Fragmentation Ratio

ILP Integer Linear Programming

ITU-T Telecommunication Sector of the International Telecommunications Union

MCF Multicore Fiber

MMF Multimodo Fiber

MTTR Mean Time to Repair

OEO Optical-Electrical-Optical

OFDM Orthogonal Frequency Division Multiplexing

OXC Optical Cross-Connect

p-cycle Pre-configured protection cycles

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QoP Quality of Protection

QoT Quality of Transmission

QPSK Quadrature Phase Shift Keying

RMLCSA Routing, Modulation Level, Spectrum and core Allocation

RMLSA Routing, Modulation Level and Spectrum Assignment

RSA Routing and Spectrum Allocation

RSCA Routing, Spectrum and Core Allocation

RWA Routing and Wavelength Assignment

SBPP Share Backup Path Protection

SCA Spectrum and Core Allocation

SDM Spatial Division Multiplexing

SRLG Shared Risk Link Group

WDM Wavelength Division Multiplexing

Lista de Algoritmos

1	Algoritmo genérico	44
2	SBPPMC	51
3	FIPPMC	52
4	MIFMC	54
5	BARTMAN	77
6	PERFECTA	78
7	MMIFMC	80
8	SBRC	101
9	STOP	102
10	FRSCA	104
11	SSSPF	105
12	RAMBO	125
13	APPROVE	127
14	MAROTO	128
15	MOFIO	151
16	PRIME	153
17	MINION	155

Sumário

1	Introdução	23
1.1	Objetivo	24
1.2	Contribuições	24
1.3	Publicações	26
1.3.1	Periódicos	26
1.3.2	Conferências	26
1.3.3	Outros	27
1.4	Organização da Tese	27
2	Conceitos Fundamentais	29
2.1	Redes ópticas	29
2.1.1	Redes ópticas WDM	30
2.1.2	Redes ópticas elásticas (EON)	30
2.1.3	Redes ópticas elásticas com multiplexação espacial	31
2.2	Sobrevivência em redes ópticas	34
2.3	Proteção na camada óptica	35
2.3.1	Proteção de enlace	35
2.3.2	Proteção de caminho	36
2.3.3	Proteção de enlace e caminho	38
2.4	<i>P-cycle</i>	38
2.4.1	Tipos de <i>P-cycle</i>	39
2.4.2	<i>P-cycle</i> FIPP	40
2.5	Roteamento multicaminho	41
2.6	Agregação de tráfego em redes ópticas	41
2.7	Multigrafo	42
2.8	Algoritmo genérico	43
2.9	Síntese do capítulo	43
3	Proteção em redes ópticas elásticas com multiplexação espacial	45
3.1	Trabalhos relacionados	46
3.2	Multigrafo	47
3.3	Notação	49
3.4	O algoritmo SBPPMC	50
3.5	O algoritmo FIPPMC	51
3.6	O algoritmo MIFMC	53
3.7	Resultados numéricos	55
3.8	Síntese do capítulo	71

4	Uso de diferentes formatos de modulação em redes ópticas elásticas com multiplexação espacial e com proteção de caminho	72
4.1	Trabalhos relacionados	74
4.2	Notação	75
4.3	O algoritmo BARTMAN	76
4.4	O algoritmo PERFECTA	78
4.5	O Algoritmo MMIFMC	79
4.6	Resultados numéricos	81
4.7	Síntese do capítulo	96
5	Agregação de tráfego e sobreposição de espectro para proteção de redes ópticas elásticas com multiplexação espacial	97
5.1	Trabalhos relacionados	98
5.2	Notação	99
5.3	O algoritmo SBRC	100
5.4	O algoritmo STOP	102
5.5	O algoritmo FRSCA	103
5.6	O algoritmo SSSPF	105
5.7	Resultados numéricos	106
5.8	Síntese do capítulo	121
6	Modulação adaptativa, agregação de tráfego e sobreposição de espectro para proteção de redes ópticas elásticas com multiplexação espacial	122
6.1	Trabalhos relacionados	123
6.2	Notação	124
6.3	O algoritmo RAMBO	124
6.4	O algoritmo APPROVE	126
6.5	O Algoritmo MAROTO	128
6.6	Resultados numéricos	130
6.7	Síntese do capítulo	146
7	Roteamento híbrido para proteção de redes ópticas elásticas com multiplexação espacial	147
7.1	Trabalhos relacionados	148
7.2	Notação	149
7.3	O algoritmo MOFIO	150
7.4	O algoritmo PRIME	151
7.5	O algoritmo MINION	153
7.6	Resultados numéricos	156
7.7	Síntese do capítulo	170
8	Considerações Finais e Perspectivas Futuras	171
8.1	Considerações Finais	171
8.2	Limitações	173
8.3	Perspectivas Futuras	173
	Referências Bibliográficas	174

A	Flexgridsim	185
A.1	Projeto do simulador Flexgridsim	185
A.1.1	Configuração do ambiente de simulação	186
A.1.2	Geração dos eventos da simulação	186
A.1.3	Simulação	187
A.1.4	Avaliação dos resultados da simulação	187
A.2	Execução do simulador Flexgridsim	187
A.3	Validação	188

Capítulo 1

Introdução

Nas últimas décadas, o tráfego de rede tem crescido a uma taxa exponencial, e não existe nenhuma indicação de que esta tendência cessará [94]. O grande problema dessa tendência é que uma fibra de único núcleo, fibra amplamente utilizada nas redes ópticas existentes, possui capacidade limitada e sua capacidade não deverá ser suficiente em um futuro próximo [24]. Estima-se que, nos próximos dez anos, o acesso do usuário final irá crescer de 100 mb/s para 1 Gb/s, motivado por inovações tecnológicas de fibras de alta capacidade para o segmento de acesso. Inevitavelmente, o núcleo da rede deverá comutar a 1 Tb/s [94], o que exige uma maior quantidade de recursos, levando, portanto, a adoção de fibras com múltiplos núcleos.

As redes ópticas elásticas (EON) com multiplexação espacial (SDM) sobre fibra multi-núcleo (MCF) são uma solução promissora para lidar com as crescentes demandas de banda passante. A capacidade de se alocar flexivelmente o espectro permitirá que estas redes lidem com demandas de tráfego com requisitos de banda variável. Além disso, a adoção de SDM irá proporcionar às redes mais capacidade. No entanto, apesar de tais vantagens, existem, ainda, poucos estudos relacionados à algoritmos de roteamento em redes ópticas elásticas SDM.

A utilização de SDM não é um conceito novo, uma vez que a utilização de vários canais espaciais (ou seja, núcleos ou modos) na mesma estrutura de fibra remete ao início dos anos 80 [4]. Contudo, apenas recentemente a abordagem de fibra multi-núcleo (MCF) tem sido explorada conjuntamente com a abordagem alternativa de usar vários modos em uma única fibra multimodo (MMF) para definir canais espacialmente distintos [83]. Ainda assim, esta tecnologia somente mostra-se possível devido a flexibilidade das redes ópticas elásticas. Essas inovações de transmissões apresentadas ao longo dos últimos anos demonstram a viabilidade de se aumentar a capacidade de transmissão em mais de uma ordem de magnitude. Para que haja a implantação dessa tecnologia é necessária a comprovação de seus benefícios no nível de rede.

Apesar da capacidade de sobrevivência ser de suma importância em redes que transportam uma grande quantidade de tráfego, a maioria dos estudos sobre EONs não considera a proteção dessas redes [111, 118, 99, 101]. As redes ópticas elásticas com multiplexação espacial sobre fibra multi-núcleo (MCF) proporcionarão maior capacidade de banda que as redes com apenas um canal, havendo, assim, uma maior necessidade de proteção, já que um corte de fibra implicará em grandes perdas de dados, maiores do que as das atuais redes ópticas.

Um dos problemas fundamentais em EON é o problema de roteamento e atribuição de espectro (RSA), que deve considerar a alocação contígua e contínua do espectro para todos os en-

laces de um caminho óptico [30]. A inclusão de multiplexação espacial incrementa o problema tradicional de RSA, que se torna o problema de roteamento e alocação de núcleo e espectro (RSCA). RSCA para redes ópticas elásticas SDM têm sido propostos, recentemente, na literatura [62], [102], [29], [30], [119], [89], [88], [90] [32], [112], [109] e [50]. Entretanto, apenas o algoritmo em [100] aborda a proteção de caminhos.

1.1 Objetivo

O objetivo desta Tese é *propor soluções de proteção em redes ópticas elásticas com multiplexação espacial, visando a redução do bloqueio de requisições para estabelecimento de conexão e o melhoramento da utilização dos recursos em redes EON-SDM, considerando diferentes cenários de carga e topologias*. Para tal, o problema de proteção destas redes levará em consideração a utilização de diferentes formatos de modulação, o uso de *agregação de tráfego*, o uso de *sobreposição de espectro em caminhos de backup*, *interferência mínima*, bem como, a *utilização de roteamento multicaminho*.

Investiga-se, primeiramente, técnicas de proteção fim-a-fim que possam eficientemente proteger caminhos ópticos, mantendo uma baixa sobrecarga da rede. As soluções propostas envolvem p-cycle FIPP, interferência mínima e caminhos de *backup* compartilhados.

Em seguida, estuda-se o problema de proteção utilizando modulação adaptativa, que leva ao aumento do número de requisições aceitas, bem como a diminuição do consumo energético. São considerados diversos formatos de modulação. Para resolver este tipo de problema são empregadas extensões dos algoritmos desenvolvidos a priori.

Esta Tese introduz, também, um estudo de agregação de tráfego e de sobreposição de espectro para proteção de redes ópticas elásticas com multiplexação espacial. O estudo lida com proteção de caminhos que fazem sobreposição do espectro seguindo restrições predefinidas.

Após a efetiva comprovação dos benefícios da utilização de modulação adaptativa e de agregação de tráfego e de sobreposição de espectro, investiga-se o uso conjunto dessas técnicas em algoritmos para proteção de redes ópticas com multiplexação espacial. Para isto, utiliza-se em conjunto as técnicas de proteção fim-a-fim (Capítulo 3), a modulação adaptativa (Capítulo 4), e a agregação de tráfego e sobreposição de espectro (Capítulo 5).

Aborda-se, também, nesta Tese, o problema de proteção em redes ópticas elásticas utilizando roteamento multicaminho. Nesse estudo, o número de caminhos necessários e a quantidade de recursos alocados para cada caminho são os dois principais parâmetros. As soluções propostas utilizam algoritmos híbridos que utilizam roteamento multicaminho quando não é possível a utilização de caminhos simples.

1.2 Contribuições

Neste trabalho são propostos dezesseis algoritmos para proteção de redes ópticas elásticas com multiplexação espacial. Os algoritmos propostos avaliam a introdução da utilização das técnicas de modulação adaptativa, agregação de tráfego, sobreposição de espectro e roteamento híbrido no bloqueio de demandas. Além disso, os algoritmos utilizam três tipos diferentes de proteção

de caminho: caminho de *backup* compartilhado, *p-cycle* FIPP e *p-cycle* FIPP com interferência mínima.

A seguir, são listadas as contribuições desta Tese para o avanço do conhecimento sobre redes ópticas elásticas utilizando multiplexação espacial:

- Um algoritmo para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando o *crosstalk* da rede e utilização de *p-cycle* FIPP.
- Um algoritmo para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando o *crosstalk* da rede, *p-cycle* FIPP e interferência mínima.
- Um algoritmo para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando o *crosstalk* da rede e caminhos de *backup* compartilhados.
- Algoritmos para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando o *crosstalk* da rede e modulação adaptativa.
- Algoritmos para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando o *crosstalk* da rede e utilizando agregação de tráfego e sobreposição de espectro.
- Algoritmos para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando de maneira conjunta o *crosstalk* da rede e utilizando agregação de tráfego, sobreposição de espectro e modulação adaptativa.
- Algoritmos para o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando o *crosstalk* da rede e utilizando roteamento híbrido, ou seja, roteamento monocaminho e multicaminho.

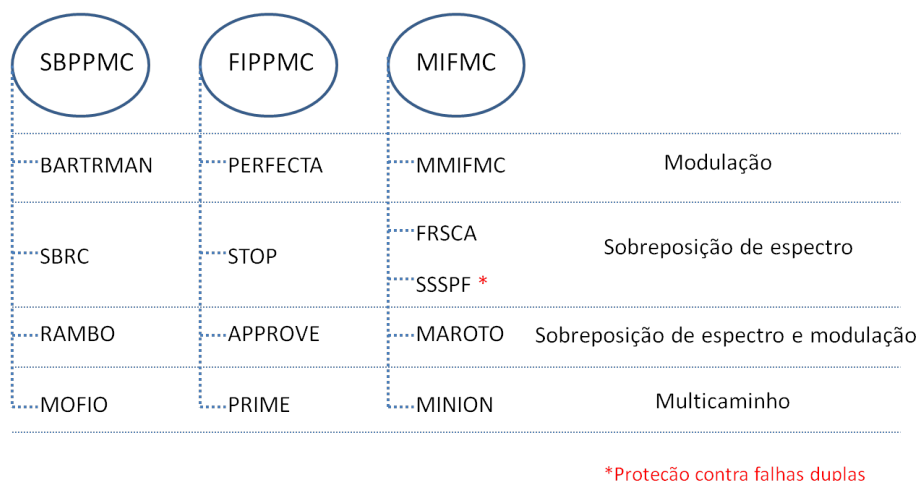


Figura 1.1: Visão geral das características dos algoritmos desenvolvidos

A Figura 1.1 apresenta uma categorização das características dos algoritmos desenvolvidos. Na primeira linha da figura, tem-se os algoritmos SBPPMC, FIPPMC e MIFMC que são os

algoritmos base no desenvolvimento desta Tese, seguidos de suas correspondentes extensões. A coluna da direita representa a extensão no algoritmo base.

O principal desafio para proporcionar proteção às redes EON-SDMs consiste na realização da escolha dos espectros a serem utilizados nos caminhos ópticos, de forma a manter a contiguidade e a continuidade de núcleo e espectro, bem como obter o menor *crossstalk* possível, reduzindo a utilização dos recursos. Técnicas como modulação, sobreposição de espectro e compartilhamento de caminhos de *backup* foram utilizadas.

1.3 Publicações

Foram documentados os resultados desta Tese nos seguintes artigos:

1.3.1 Periódicos

1. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Algoritmo de p-Cycle e Interferência Mínima para Proteção de Redes Ópticas Elásticas com Multiplexação Espacial," in IEEE Latin America Transactions, vol. 15, no. 7, 2017 [73].
2. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Protection, Routing, Modulation, Core and Spectrum Allocation in SDM Elastic Optical Networks" in IEEE Communications Letters, 2018 [66].
3. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Routing, Spectrum and Core Assignment Algorithms for Protection of Space Division Multiplexing Elastic Optical Networks" in Journal of Network and Computer Applications, 2018 (Submetido).

1.3.2 Conferências

1. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "The Minimum Interference p-Cycle Algorithm for Protection of Space Division Multiplexing Elastic Optical Networks," in IEEE Latin-American Conference on Communications, 2016 [71].
2. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Algorithm for Protection of Space Division Multiplexing Elastic Optical Networks," in IEEE Global Communications Conference, 2016 [70].
3. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Algorithm for Shared Path for Protection of Space Division Multiplexing Elastic Optical Networks," in IEEE International Conference on Communications, 2016 [65].
4. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Proteção de Redes Ópticas Elásticas com Multiplexação Espacial Baseada em Modulação, p-Cycle FIPP e Interferência Mínima," in XXXV Brazilian Symposium on Computer Networks and Distributed Systems 2017 [74].

5. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Sharing Spectrum and Straddling p-Cycle FIPP for Protection Against Two Simultaneous Failures in SDM Elastic Optical Networks," in IEEE Latin-American Conference on Communications, 2017 [76].
6. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Routing, Spectrum, Core and Modulation Level Assignment Algorithm for Protected SDM Optical Networks," in IEEE Global Communications Conference, 2017 [75].
7. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Spectrum Overlap and Traffic Grooming in P-cycle Algorithm Protected SDM Optical Networks," in IEEE International Conference on Communications, 2018 [67].
8. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da, "Proteção de Redes Ópticas Elásticas com Multiplexação Espacial Baseada em Agregação de Tráfego, Sobreposição de Espectro e Compartilhamento de Caminho de Backup," in XXXVI Brazilian Symposium on Computer Networks and Distributed Systems 2018 [77].
9. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da, "P-cycle Protected Multipath Routing, Spectrum and Core Allocation in SDM Elastic Optical Networks," in IEEE International Conference on Communications, 2019 (Submetido).

1.3.3 Outros

Os trabalhos abaixo, apesar de não fazerem parte deste texto, foram produzidos durante o doutorado e contribuíram de forma indireta para o conteúdo desta Tese, uma vez que estes não tratam multiplexação espacial.

1. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da. Proteção de Redes Ópticas Elásticas Baseada em Agregação de Tráfego, Sobreposição de Espectro e p-Cycle FIPP. In Proc of XXXIV Brazilian Symposium on Computer Networks and Distributed Systems, p 1-14, 2016 [72].
2. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Traffic Grooming and Spectrum Overlap in FIPP p-cycle for Protection of Elastic Optical networks," in IEEE Latin-American Conference on Communications, 2016 [64].
3. OLIVEIRA, H. M. N. S.; Fonseca, N. L. S. da , "Protection in Elastic Optical Networks Using Failure-Independent Path Protecting p-Cycles" in Optical Switching and Networking, 2018 (Submetido).

1.4 Organização da Tese

Esta Tese apresenta o desenvolvimento de soluções eficientes para proteção em redes ópticas elásticas com multiplexação espacial e está organizada da seguinte forma:

O Capítulo 2 revisa conceitos em redes ópticas e suas variações, o problema de sobrevivência em redes ópticas, o problema de proteção na camada óptica, a técnica *p-cycle*, os problemas

de roteamento multicaminho e agregação de tráfego, o multigrafo e por fim, apresenta um algoritmo para resolver de forma genérica o problema de proteção de redes ópticas elásticas com multiplexação espacial.

O Capítulo 3 explora, por sua vez, o problema de proteção em redes ópticas elásticas com multiplexação espacial, considerando a minimização da probabilidade de bloqueio de demandas que chegam à rede, bem como, a melhora da utilização dos recursos disponíveis considerando diferentes tipos de proteção de caminho.

O Capítulo 4 avalia o impacto da utilização de modulação adaptativa no bloqueio de demandas em redes ópticas elásticas com multiplexação espacial. Para isso, escolhe-se para cada caminho óptico diversos formatos de modulação para ser utilizado em conjunto com técnicas de proteção de caminho. A utilização de modulação adaptativa além de reduzir o bloqueio de banda provê um aumento na eficiência energética.

O Capítulo 5 analisa o impacto da proteção de redes ópticas elásticas com multiplexação espacial através da utilização de agregação de tráfego e sobreposição de espectro em técnicas de proteção de caminho. O uso de agregação de tráfego e sobreposição de espectro reduz a utilização de recursos, consequentemente reduzindo o bloqueio de demandas. A agregação de tráfego permite que os comutadores agrupem múltiplos caminhos ópticos em um único transmissor o que reduz a necessidade de bandas de guarda. A sobreposição de espectro possibilita que caminhos ópticos de backup compartilhem a reserva de *slots*.

O Capítulo 6 investiga o impacto de se utilizar de forma conjunta a modulação adaptativa, agregação de tráfego e sobreposição de espectro. Enquanto a modulação adaptativa permite o ajuste flexível da largura de banda, a agregação de tráfego e sobreposição de espectro diminuem a quantidade de recursos utilizados, diminuindo, ainda mais, o bloqueio de requisições.

O Capítulo 7 compara a proteção em redes ópticas elásticas com multiplexação espacial utilizando roteamento monocaminho e roteamento híbrido (monocaminho e multicaminho). Os algoritmos híbridos propostos utilizam proteção de caminho e priorizam sempre que possível a utilização de roteamento simples. A utilização de multicaminho distribui o tráfego, provendo um melhor balanceamento na utilização de recursos da rede e o aumento na vazão da rede.

Por fim, as considerações finais e as perspectivas futuras são apresentadas no Capítulo 8.

Capítulo 2

Conceitos Fundamentais

Neste capítulo, são apresentados alguns conceitos básicos sobre os assuntos envolvidos nesta Tese. Na Seção 2.1, apresenta-se o conceito de redes ópticas e suas variações. Na Seção 2.2, descreve-se o conceito de sobrevivência em redes ópticas. Na Seção 2.3, apresentam-se esquemas de proteção em redes ópticas. Na seção 2.4, introduz-se o problema de roteamento multicaminho. Por fim, na seção 2.5, descreve-se o problema de agregação de tráfego.

2.1 Redes ópticas

As redes ópticas são redes de comunicação de dados que operam em altíssima velocidade e utilizam a luz para transmissão de dados. Estas redes possuem baixa latência e permitem o uso de grande quantidade de banda passante, permitindo a expansão da Internet. Essas redes oferecem banda passante segura e dedicada. Contudo, são ineficientes na utilização dos recursos da rede, que ficam ociosos quando um usuário não transmite dados.

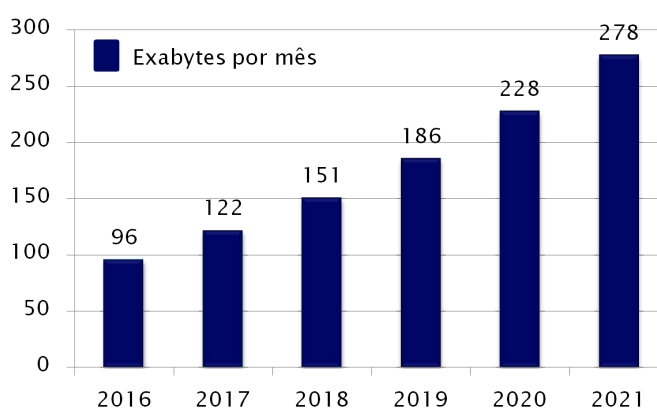


Figura 2.1: Previsão do crescimento de tráfego na Internet
Fonte: Cisco Visual Networking Index

A evolução das fibras ópticas teve seu marco inicial com o desenvolvimento da tecnologia WDM (*Wavelength Division Multiplexing*). Nessa tecnologia, a largura de banda de uma fibra

óptica é dividida em um certo número de comprimentos de ondas, cada um dos quais possui alta taxa de transmissão. Com o intuito de diminuir os recursos ociosos decorrentes do uso da grade fixa das redes WDM, as redes ópticas elásticas foram propostas. Sua principal vantagem é a capacidade de transmitir dados em diferentes taxas, alocando o espectro óptico de acordo com a demanda, garantindo, assim, uma alta eficiência espectral. No entanto, a capacidade de roteamento sob fibras ópticas de único núcleo esta chegando ao seu limite (ver Figura 2.1), o que tem motivado a exploração da utilização de multiplexação espacial (SDM). As próximas subseções descrevem com mais detalhes as tecnologias citadas.

2.1.1 Redes ópticas WDM

A tecnologia WDM proporciona a multiplexação por comprimento de onda, o que aumenta consideravelmente a capacidade de transmissão em uma fibra. Para o estabelecimento de uma comunicação entre um par de nós de uma rede óptica WDM é necessário construir um caminho óptico fim-a-fim. Cada caminho óptico consiste em uma rota fixa e um comprimento de onda. O problema da escolha de uma rota e de um comprimento de onda adequado para cada caminho óptico é conhecido como roteamento e alocação de comprimento de onda, *Routing and Wavelength Assignment (RWA)*.

A tecnologia WDM permite a existência de uma grande quantidade de canais em uma única fibra óptica. Cada canal opera a velocidade de Gbit/s, sendo que a capacidade total de um enlace WDM pode chegar a taxas de transmissão da ordem de Tbit/s[63]. As redes WDM empregam alocação de espectro em unidades de tamanho fixo, denominadas de comprimento de onda, e faixas de espectro chamadas de banda de guarda, para separar dois comprimentos de ondas adjacentes. O provisionamento usando largura de banda fixa de um comprimento de onda leva à subutilização do espectro uma vez que a demanda dos fluxos de dados raramente coincidirão com a largura de banda exata de um comprimento de onda [63].

As redes ópticas WDM com capacidade de agregação permitem que vários fluxos de tráfego compartilhem a banda passante de um caminho óptico. No entanto, esta técnica não elimina o desperdício de recursos e demanda gerenciamento em tempo real de eventuais agregações de tráfego. Esse problema é denominado de agregação de tráfego. Tal rigidez motivou recentemente o surgimento das redes ópticas elásticas.

2.1.2 Redes ópticas elásticas (EON)

Nos últimos anos, tem crescido a busca por uma tecnologia de rede óptica que não usa a grade fixa de comprimentos de onda. Nestas redes, aloca-se largura de banda com granularidade fina. O conceito de redes ópticas elásticas foi introduzido em *Jinno et.al* [48] e estas redes são também chamadas, na literatura, de redes flexíveis. Nelas, o espectro da fibra é tratado como um recurso "contínuo", possibilitando seu uso de forma mais eficiente. Basicamente, o caminho óptico pode ser alocado em faixas estreitas (*slots* tipicamente, de 12,5 GHz) do espectro separados por uma banda de guarda, denominada de *Filter Guard Band (FGB)*.

Dada a introdução de uma nova forma do uso do espectro, novos tipos de roteamento, gerenciamento, agrupamento de tráfego, restauração e proteção devem ser redefinidos para essa nova tecnologia. Para as redes ópticas elásticas, algoritmos de roteamento e atribuição de com-

primentos de ondas (RWA), utilizado em redes WDM, não são mais aplicáveis, necessitando-se de novos algoritmos denominados, Roteamento e alocação do espectro, *Routing and Spectrum Allocation* (RSA). O RSA consiste em encontrar rotas e banda do espectro para um par de nós origem e destino com a finalidade de estabelecer uma conexão. Essa alocação está sujeita às restrições de continuidade e contiguidade do espectro. A primeira impõe que o conjunto de *slots* atribuídos por um algoritmo deve ser o mesmo em todos os enlaces ao longo de uma rota para evitar a conversão opto-elétrica do sinal óptico, enquanto a segunda impõe que os *slots* alocados devem ser contíguos, para alcançar alta eficiência espectral.

Buscando melhorar a utilização do espectro nas redes ópticas elásticas, diferentes formatos de modulação podem ser utilizados para alcançar maiores taxas de transmissão. No entanto, a transmissão de um elevado número de bits por símbolo depende do comprimento dos enlaces do caminho, uma vez que a distância impacta a capacidade do receptor de decodificar o sinal recebido [60]. Tal restrição impõe outro componente para o problema *RSA*, a escolha do melhor nível de modulação a ser usado ao longo do caminho óptico. Esse problema é chamado de roteamento, nível de modulação e atribuição do espectro, *Routing, Modulation Level and Spectrum Assignment* (RMLSA).

2.1.3 Redes ópticas elásticas com multiplexação espacial

As redes ópticas elástica (EON) convencionais utilizam as dimensões de tempo e frequência para fornecer recursos de redes ópticas com demanda de tráfego variável. Pesquisas atuais consideram a exploração de uma terceira dimensão, o espaço. A adição da dimensão do espaço é realizada através da utilização de múltiplos núcleos de fibra.

MCFs são uma forma eficiente de concepção e fabricação para utilização de multiplexação espacial (SDM), melhorando, significativamente, o potencial da capacidade de transmissões futuras e a eficiência espectral das redes [80]. A adoção destas fibras ópticas que suportam multiplexação espacial aumenta a capacidade das fibras por um fator C , em que C é o número de núcleos espaciais dentro da fibra.

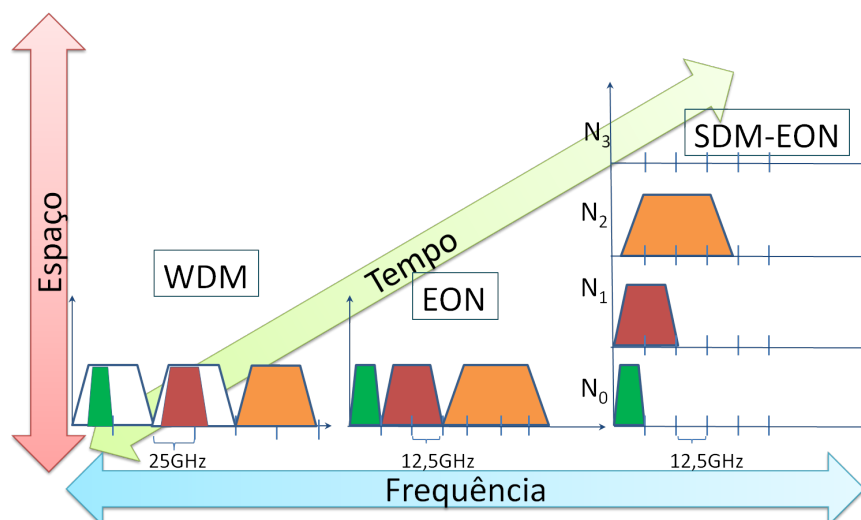


Figura 2.2: Três possibilidades diferentes para provisionamento de recursos.

A Figura 2.2 ilustra a alocação de espectro em diferentes tecnologias de redes. Na figura, para rede EON-SDM, assume-se que existe um sistema SDM com quatro núcleos, além disso, desconsidera-se a utilização de banda de guarda. As rede WDMs utilizam menos eficientemente os recursos do que as redes EONs, visto que nas redes WDMs, um comprimento de onda acomoda diferentes demandas, subutilizando a largura de banda disponível. Em uma rede óptica elástica com multiplexação espacial, se a rede precisa de uma conexão de 1 Tb/s entre dois nós, a flexibilidade proporcionada pelo uso de três dimensões (temporal, espectral e espacial) permite que esta conexão de 1 Tb/s seja estabelecida por diferentes núcleos. Desconsiderando a dimensão espacial, ou seja, utilizando apenas um núcleo, tem-se uma menor flexibilidade.

Além dos efeitos físicos presentes em EON (acumulação de ruído óptico, mistura de quatro ondas), a dimensão espacial pode introduzir uma quantidade significativa de interferência de banda e *crosstalk* (XT) nos dispositivos multiplexadores/demultiplexadores (MUX/DEMUX) espaciais distribuídos ao longo dos enlaces de fibra, devido ao modo de acoplamento utilizado.

Em redes WDM e EON, quando uma requisição de largura de banda chega a um nó, a rede tem de alocar uma parte do espectro (alocação de recursos) para esta requisição e encontrar um caminho físico disponível para o nó destino (roteamento). Com a utilização de multiplexação espacial, esta operação torna-se mais flexível, porém, aumenta a complexidade de encaminhamento e alocação. Este problema é denominado de Roteamento e alocação de núcleo e espectro, (*Routing, spectrum and core allocation*, RSCA). Os algoritmos RSCA alocam rotas e banda do espectro como os algoritmos RSA, mas também alocam núcleos para o estabelecimento de uma conexão. A adição da dimensão espacial na multiplexação aumenta a complexidade da atribuição do espectro, uma vez que o espectro de todos os núcleos deve ser levado em consideração durante o cálculo das rotas. Além disso, soluções que gerenciam o roteamento em redes ópticas com multiplexação espacial devem alocar a largura de banda disponível, levando em consideração o *crosstalk* entre núcleos [29], [62] e [61].

Atualmente as abordagens de SDM para fibra óptica variam para diferentes tecnologias, tais como: *Few-mode fibers* (FMFs), *Multimode fibers* (MMF). *Multi-core fibers* (MCFs), *Coupled-core fibers* (CCFs).

A Figura 2.3 ilustra esquemas de representação para redes ópticas com multiplexação espacial.

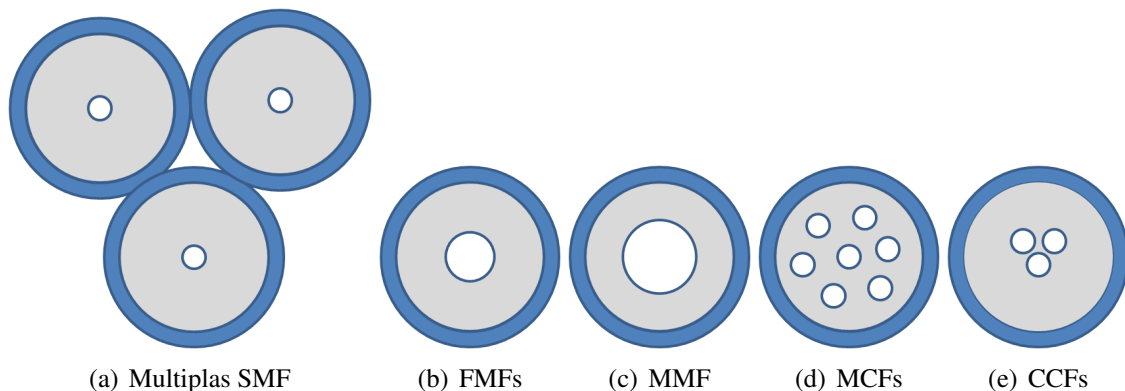


Figura 2.3: Esquemas de representação para redes ópticas SDM

Os requisitos para uma fibra com poucos modos são semelhantes aos das fibras de transmissão monomodo: baixa atenuação, pequena não linearidade e um elevado coeficiente de dispersão. Além disso, as fibras devem suportar um número específico de modos e baixo acoplamento entre os modos [37, 95, 56].

As fibras multimodo seguem a recomendação ITU-T [46]. Nela, é descrito que um cabo de fibra óptica multimodal possui índice graduado de 50/125 μm . Em MMF, o número de modos suportados por uma fibra depende do tamanho do núcleo e do índice de refração do revestimento da fibra.

Takara *et.al.* [97, 44] realizaram experimentos utilizando fibras de múltiplos núcleos e demonstraram a grande capacidade de transmissão dessas fibras. Demonstraram, também, que o aumento da capacidade de transmissão de MCF, aumenta a eficiência espectral, definida como um produto do número de núcleos e da eficiência espectral por núcleo. O *crosstalk* também é analisado. Em MCF, cada núcleo atua como uma fibra de único modo.

A fibras com núcleos acoplados [85] consistem em 3 núcleos de modo único uniformemente espaçados em um rebordo padrão de 125 μm de diâmetro.

Atualmente, o uso mais adequado de multiplexação espacial não é definido. A escolha da melhor solução não depende apenas do meio de fibra, mas também de componentes ópticos como (de) multiplexadores, amplificadores, filtros ópticos e *switches*. Existem duas abordagens para subdividir os tipos de fibras acima, nomeadamente multimodo e múltiplos núcleos, na qual os modos de fibra ortogonal ou núcleos espaciais ortogonais são utilizados como canais de transmissão. O primeiro tipo de fibra compreende o FMF e MMF, e o último tipo de fibra compreende CCFs e MCFs.

Para EON-SDM, além da tecnologia utilizada, a forma de alocação também pode variar. De acordo com a forma de alocação, estas redes podem suportar supercanais, que utilizam mais de um núcleo para atender uma única requisição, com isto, é desrespeitada a restrição de contiguidade do espectro, o que implica na necessidade de banda guarda para separar cada conjunto de espectro em cada núcleo, podendo implicar em um maior *crosstalk* entre núcleos. Outra forma de alocação é a utilização de diferentes núcleos para o mesmo caminho, ou seja, nesta abordagem não é considerada a restrição de continuidade do núcleo. Um caminho pode utilizar núcleos diferentes para enlaces diferentes, gerando um aumento na complexidade e no custo dos OXCs utilizados. Por fim, uma outra forma de alocação é a utilização de apenas um núcleo por caminho, mantendo a contiguidade e continuidade de núcleo e espectro. Nesta forma de alocação, os OXCs possuem menor complexidade e apenas a necessidade de uso de banda de guarda entre conexões diferentes.

Neste trabalho, consideramos a abordagem MCFs devido às vantagens de geração de menor *crosstalk*, independência entre núcleos e maior custo benefício. Além disso, a alocação de recursos considera os OXCs mais simples, em que é mantida a contiguidade e continuidade de núcleo e espectro.

Crosstalk

No contexto de redes ópticas com multiplexação espacial, o *crosstalk* (XT) é a interferência que um núcleo de um enlace de transmissão causa em outro núcleo do mesmo enlace, ou seja, a relação entre a potência óptica inserida pelos núcleos adjacentes e em um núcleo específico,

dividida pela potência do sinal nesse núcleo e medida em *dB*. O limite além do qual a integridade do sinal é alterada pode variar entre -16 dB e -32 dB [106], dependendo do formato de modulação usado [91, 106].

O uso de fibras com múltiplos núcleos para multiplexação espacial tem motivado diversas investigações sobre questões como: alocação eficiente dos núcleos para reduzir o *crosstalk* e definição de características para espessura do revestimento [41, 45, 86, 91, 98]. Obviamente, o desafio é juntar tantos núcleos quanto possível em uma única estrutura de fibra, evitando grandes penalidades de interferências *inter core*.

Os algoritmos RSCA precisam assegurar que a condição do espectro alocado a uma conexão seja tal que o sinal transmitido possa ser decodificado no destino. Portanto, antes de alocar uma região do espectro para uma conexão, o nível de *crosstalk* nesta região precisa ser verificado. Se estiver acima de um valor limite inaceitável, a região não deve ser alocada. Além disso, mesmo se o nível de *crosstalk* estiver abaixo do valor limite, a contribuição do impacto do *crosstalk* gerado pela conexão de entrada no *crosstalk* de todas as conexões precisa ser avaliada. Se tal incremento em *crosstalk* levar a um valor de *crosstalk* maior que o limiar, a conexão não deve ser aceita, pois degradará a qualidade de transmissão em conexões já estabelecidas.

Para calcular o *crosstalk* (*XT*) de um núcleo com n núcleos vizinhos ativos de um MCF homogêneo, utilizamos uma fórmula baseada na Eq. 2.1 como em [41, 42, 91], que também considera a teoria do poder acoplado [52], levando à Eq. 2.2. A Equação 2.2 será utilizada nesta tese para calcular o *crosstalk* nos cálculos para garantir a disponibilidade das conexões com relação aos núcleos vizinhos.

$$h = \frac{2 \cdot k^2 \cdot R}{\beta \cdot D} \quad (2.1)$$

Na Eq. 2.1, h é o aumento da interferência média por unidade de comprimento, calculado por vários parâmetros da fibra: k , β , R , D são, respectivamente, o coeficiente de acoplamento, a constante de propagação, o raio de curvatura e a distância entre os núcleos.

$$XT = \frac{n\{1 - \exp(-(n+1) \cdot 2 \cdot h \cdot L)\}}{1 + n\{\exp(-(n+1) \cdot 2 \cdot h \cdot L)\}} \quad (2.2)$$

A Eq. 2.2 faz uso de h da equação 2.1 multiplicada por L que é o comprimento da fibra, enquanto n representa o número de núcleos adjacentes.

Nesta Tese, assume-se um valor máximo de n igual a 6. Além disso, os outros parâmetros atribuí-se valores realísticos, tais como $k = 2 \times 10^{-5}$, $R = 50 \text{ mm}$, $\beta = 4 \times 10^6$ e $D = 45 \mu\text{m}$. O pior caso de *crosstalk* para os núcleos sempre será o núcleo central (ou qualquer outro núcleo que tenha o maior número de núcleos vizinhos), uma vez que recebe interferência indesejada de todos os seus núcleos adjacentes. Enfatiza-se que o *crosstalk* entre núcleos ocorre apenas entre os mesmos *slots* de frequência usadas em núcleos adjacentes. Nas simulações, assume-se que o espectro de cada núcleo é totalmente utilizado, com exceção da banda de guarda.

2.2 Sobrevivência em redes ópticas

Em redes ópticas, a falha de um elemento de rede (por exemplo, um corte de fibra) pode causar enorme perdas de dados, dado às altas taxas de transmissão que elas possuem. Sobrevivência é

a capacidade de uma rede fornecer serviços durante falhas. Em redes ópticas, as falhas podem ser de dois tipos, falha de componentes e cortes de fibra. As taxas de falha de transmissores e receptores são 10.867 FIT e 4311 FIT, respectivamente [117, 1], enquanto que a taxa de falha devido ao corte de fibra é de 3 cortes para cada 1610 km de fibras por ano, cerca de um corte de fibra a cada quatro dias para uma rede com 50000 mil km de fibra [36]. Os tempos médios de reparo (MTTR) são 2 horas para equipamentos e 12 horas para corte de cabos [1]. As redes são, portanto, afetadas centenas a milhares de vezes principalmente devido a cortes de cabos, ou seja, enlaces falhos [2]. Nesta seção, vamos fornecer uma visão geral dos vários esquemas de recuperação de falha, enfatizando a capacidade de sobrevivência durante falhas individuais. Os esquemas de recuperação de falhas de rede podem ser classificados como proteção e restauração.

Esquemas de proteção definem previamente os recursos que são usados para *backup* [15]. Normalmente, o mesmo protocolo de roteamento é usado depois de configurar o caminho primário para determinar o caminho de *backup* (caminho secundário ou caminho alternativo). O caminho primário é utilizado para transmitir os dados e o caminho de *backup* é reservado para uso em caso de falha. Após a detecção da falha, os comutadores são re-configurados para usar o caminho de *backup*. Este esquema pode fornecer proteção garantida, já que a demanda criada será completa se e somente se o caminho de *backup* também estiver disponível. Em segundo lugar, estes sistemas podem fornecer recuperação rápida porque a computação do caminho de *backup* é realizada em conjunto com a do caminho primário. No entanto, a utilização de recursos não é eficiente devido à reserva de recursos para os caminhos de *backup*.

Esquemas de restauração referem-se à recuperação dinâmica após o aparecimento de falhas [15]. A restauração envolve a detecção de uma falha, os novos cálculos de caminho para as conexões com falha e reconfiguração de comutadores para o caminho de restauração. Esses sistemas fornecem a utilização eficiente dos recursos. No entanto, a recuperação bem sucedida não pode ser garantida, uma vez que recursos suficientes podem não estar disponíveis para recuperação no momento da falha. Além disso, o tempo necessário para a recuperação é maior. A camada óptica pode proporcionar proteção rápida enquanto que as camadas mais elevadas podem ser usadas para a restauração inteligente.

2.3 Proteção na camada óptica

Os esquemas de proteção de camada óptica podem ser divididos em proteção de caminho, proteção de enlace e proteção de enlace e caminho. Estes esquemas podem ser implementados de forma estática, bem como dinâmica. No caso estático, as demandas de tráfego são fixas e, no caso dinâmico, as demandas de tráfego mudam e os caminhos são estabelecidos e removidos de acordo com as demandas do tráfego.

2.3.1 Proteção de enlace

O mecanismo de proteção de enlace é projetado para fornecer proteção automática de comutação (APS) de anéis nas redes de malha óptica. Este mecanismo de proteção é rápido, distribuído e automático. Na proteção de enlace, o caminho óptico pode ser considerado como sendo diferentes enlaces fim-a-fim. Em caso de uma falha de enlace, um caminho de *backup* substitui

apenas o enlace com defeito, os outros enlaces do caminho óptico permanecem iguais (ver Figura 2.4).

Na Figura 2.4, o caminho em destaque AED representa o caminho primário e os caminhos AFE e EFD mostram os caminhos que protegem os enlaces AE e ED do caminho primário, respectivamente. Em caso de falha de um único enlace o caminho primário alterna apenas o caminho de *backup* do enlace, mantendo os outros enlaces do caminho.

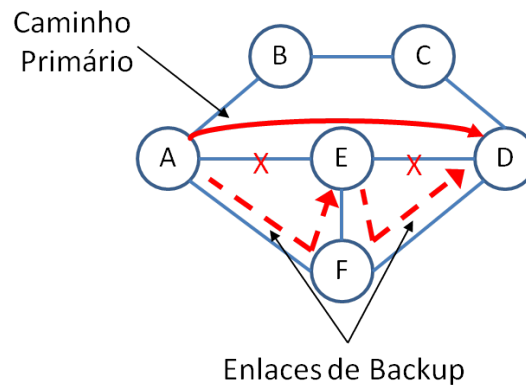


Figura 2.4: Exemplo de proteção de enlace.

Devido ao menor tamanho do enlace em relação ao caminho, esquemas de proteção de enlace são mais rápidos em resposta. No entanto, devido à recuperação local o número total de saltos pode ser maior e a utilização de recursos pode ser menos eficiente. Alguns tipos de proteção de enlaces são:

- A proteção de enlace em anel, divide a rede em ciclos menores (anéis), de forma que cada enlace fique protegido por ao menos um ciclo.
- Na proteção de enlace com duplos anéis, cada enlace da rede possui um enlace bidirecional para o caminho primário e um enlace bidirecional para o caminho de backup, ou seja, cada enlace é protegido por exatamente dois anéis.
- O *p-cycle* é um esquema de proteção em que a capacidade reservada é pré-conectada e forma estruturas em anel para proteção de redes em malha [3]. *P-cycles* fornecem proteção semelhante a proteção fornecida por *Bidirectional Line Switched Ring*, que é considerado uma generalização do esquema de proteção 1:1, ou seja, um anel protegendo um enlace.

2.3.2 Proteção de caminho

Neste esquema de proteção, todo caminho óptico é protegido. No caso de falha, a localização da falha não é necessária. Ao invés disso, o tráfego é comutado para enlaces e nós disjuntos do caminho primário. Estes esquemas são indicados para as redes ópticas, em que os nós intermediários podem não ter a capacidade de monitoramento. Os esquemas de proteção de caminho são : 1+1 (duplicado), 1:1 (dedicado), 1:N (compartilhado).

A Figura 2.5 mostra alguns exemplos de caminho de *backup*. Na Figura 2.5(a), o caminho AED representa o caminho primário e o caminho AFD representa o caminho de *backup* 1+1. No

exemplo, o nó fonte A transmite tanto no caminho primário quanto no caminho de *backup* para o nó destino D. Na Figura 2.5(b), o caminho AED representa o caminho primário e o caminho AFD representa o caminho de *backup* 1:1. No exemplo, o nó fonte A transmite no caminho primário para o nó destino D e ao mesmo tempo o caminho de backup AFD é reservado em caso de falha. Na Figura 2.5(c), os caminhos AED e ABCD representam os caminhos primários, e o caminho AFD representa o caminho de *backup* 1:N. No exemplo, o nó fonte A transmite nos caminhos primários para o nó destino D e ao mesmo tempo o caminho de *backup* AFD é reservado para ambos em caso de falha.

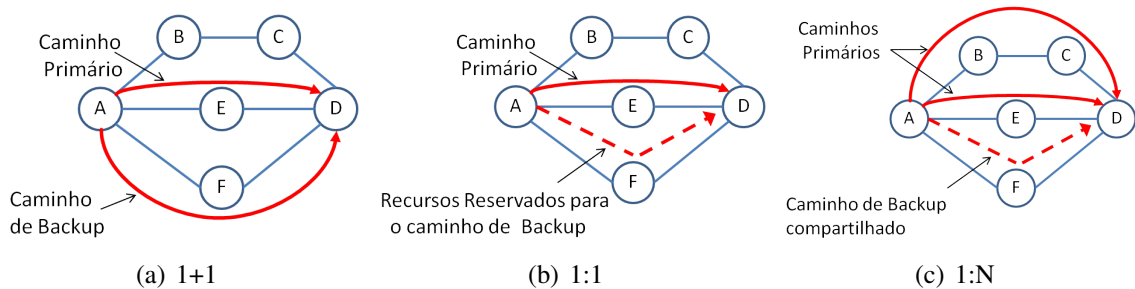


Figura 2.5: Exemplos de proteção de caminho.

- No caminho de *backup* 1+1, o nó fonte transmite tanto no caminho primário quanto no de *backup*. Esta abordagem é simples e rápida, no entanto faz grande uso dos recursos da rede.
- No caminho de backup 1:1, o nó fonte reserva recursos tanto no caminho primário quanto no de *backup*. Nele, os recursos são reservados para apenas um caminho óptico e não podem ser compartilhados. No entanto, os recursos reservados podem ser usados para transportar tráfego com baixa prioridade. Em caso de falha, o tráfego de baixa prioridade, transmitido através do caminho de *backup*, é descartado e o tráfego será protegido e direcionado através de caminho de *backup*. Os esquemas de proteção 1+1 e 1:1 foram comparados em Li *et. al.*[53].
- No caminho de backup 1:N, o caminho de *backup* é compartilhado. No momento da configuração da conexão, recursos do caminho de *backup* também são computados e reservados. Os recursos reservados podem ser usados para proporcionar proteção para outros caminhos primários. No entanto, os caminhos que são protegidos pelo caminho de *backup* devem ser disjuntos. Assim, uma única falha não afeta mais de um caminho óptico. Existem várias formas de caminhos de proteção compartilhado. Entre elas:
 - Grupo de enlaces de risco compartilhado (SRLG) é um grupo de caminhos ópticos que possui um elemento comum, cuja falha pode levar a falha de todos os caminhos ópticos do grupo[108]. Por exemplo, se uma fibra é cortada, todos os caminhos ópticos através desta fibra serão afetados, ou seja, todos os caminhos ópticos utilizando esta fibra pertencem a um SRLG.
 - Proteção de caminhos de *backup* compartilhados (SBPP) é um método de proteção em que um único caminho de *backup* pode proteger vários caminhos primários,

desde que estes sejam disjuntos entre si. O problema SBPP é NP-completo, e heurísticas têm sido desenvolvidas para encontrar os caminhos primários e de *backup* no ambiente dinâmico [78].

- *p-cycle* FIPP (caminho de proteção independente de falhas) é uma extensão do conceito de *p-cycle* que permite que falhas não sejam necessariamente limitadas a um enlace ou segmento de caminho imediatamente adjacente aos nós finais. *P-cycle* FIPP será mais detalhado na seção 2.4.2.

2.3.3 Proteção de enlace e caminho

Neste esquema, ao invés de um caminho ou um enlace, um segmento do caminho será protegido. Cada segmento terá o seu próprio caminho de *backup* e, em caso de falha em um segmento, o tráfego é comutado para o caminho de *backup* do segmento. Este esquema fornece maior eficiência do que proteção de enlace, visto que segmentos podem ser compartilhados, e como segmentos são mais curtos do que caminhos, a restauração é mais rápida.

A Figura 2.6 mostra um exemplo de proteção de segmento. Na Figura, o caminho AFD representa o caminho primário e o caminho FED representa o caminho de proteção de segmento. No exemplo, o nó fonte A transmite, no caminho primário, para o nó destino D. Contudo, o caminho de proteção FED protege apenas uma parte do caminho.

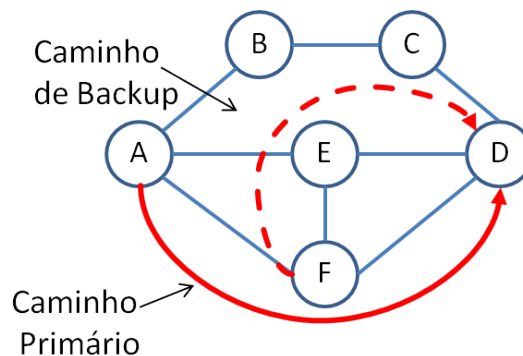


Figura 2.6: Exemplo de proteção de segmento.

A vantagem deste tipo de proteção é que o processo de computação é totalmente distribuído, tempo de recuperação reduzido, eficiência de capacidade e escalabilidade. No entanto, uma grande desvantagem é o aumento na complexidade de sinalização.

2.4 *P-cycle*

P-cycle (ciclos pré-configurados) é um tipo de ciclo usado em uma das técnicas mais promissoras para proteção de enlaces compartilhados. Os *p-cycles* são estruturas fechadas e pré-conectadas. Um *p-cycle* pode proteger todos os enlaces no ciclo, assim como os enlaces transzonais. Um enlace no ciclo são enlaces que são utilizados na criação do *p-cycle*, enquanto enlaces transzonais são aqueles que possuem seus nós finais no *p-cycle*, contudo não são utilizados pelo *p-cycle*. O *p-cycle* combina a vantagem de eficiência em proteção das redes em malha com a

vantagem da velocidade de recuperação das redes em anéis. A eficiência semelhante as redes em malha pode ser alcançada devido à proteção compartilhada fornecida pelo p -cycle a todas os enlaces no ciclo, bem como os enlaces transzonais. A velocidade de recuperação é alcançada pois a capacidade reservada da rede é pré-conectada para formar os p -cycles. Portanto, apenas duas ações de comutação nos nós finais da extensão com falha, são necessárias no caso de falha para alternar o tráfego para o caminho de proteção fornecido pelo p -cycle pré-configurado. Assim, eles qualificam ambos os requisitos; velocidade de restauração e eficiência. A diferença fundamental entre p -cycle e proteção em anel é a proteção de enlaces transzonais, que são enlaces que não estão no anel (ciclo) e cujos dois nós finais estão no ciclo. Esta propriedade melhora a eficiência de proteção. A Figura 2.7 apresenta um exemplo de enlace transzonal ao p -cycle e enlace no p -cycle. Nela, o círculo tracejado BCEF representa o p -cycle, o enlace CD representa um enlace transzonal ao p -cycle e o enlace BD representa um enlace no p -cycle.

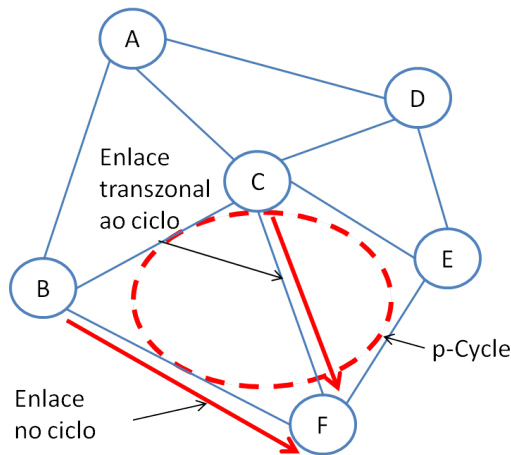


Figura 2.7: Exemplo de enlace transzonal e no ciclo.

2.4.1 Tipos de P -cycle

Os vários p -cycles discutidos na literatura foram nomeados com base na sua relação estrutural com a rede e o tipo de proteção fornecida por eles.

De acordo com a relação estrutural eles podem ser classificados como [3]:

- P -cycle hamiltoniano (*Hamiltonian p-cycle*) - p -cycle que passa por todos os nós da rede apenas uma vez.
- P -cycle simples (*simple p-cycle*) - p -cycle que não passa por qualquer nó ou extensão mais de uma vez.
- P -cycle não simples (*non-simple p-cycle*) - p -cycle que passa através de um nó por mais de uma vez.

De acordo com relação de proteção, eles podem ser classificados como [3]:

- P -cycle ao redor do nó (*node encircling p-cycle*): passa através dos nós adjacentes do nó que falhou, mas não através do nó falho em si.

- *P-cycle* de enlaces (*span p-cycle*) - o *p-cycle* simples que fornece proteção de enlace.
- *P-cycle* de fluxo (*flow p-cycle*) - fornece proteção para o segmento de caminho situado no *p-cycle*.
- *P-cycle* de proteção de caminho com independência de falha (*Failure Independent Path Protection p-cycle*)- pode proteger o caminho completo, tendo o nó fonte e o nó de destino situado no *p-cycle*.

2.4.2 *P-cycle* FIPP

Nesta Tese, ênfase será dada ao *p-cycle* de Proteção de Caminho com Independência de Falha (FIPP), proposto em [51]. *p-Cycles* FIPP fornecem proteção de caminhos para caminhos com nós finais sobre o *p-cycle*. FIPPs proporcionam a vantagem da detecção de falha independente da localização da falha, ou seja são “independente de falha”. Tal propriedade é vantajosa quando a localização da falha é lenta ou difícil, como em redes transparentes e translúcidas. Além do que *p-cycle* FIPP é capaz de atingir a velocidade de restauração rápida mantendo a eficiência de espectro [51]. A técnica *p-cycle* FIPP mostrou-se promissora para proteção de redes ópticas elásticas [102], [68] e [13].

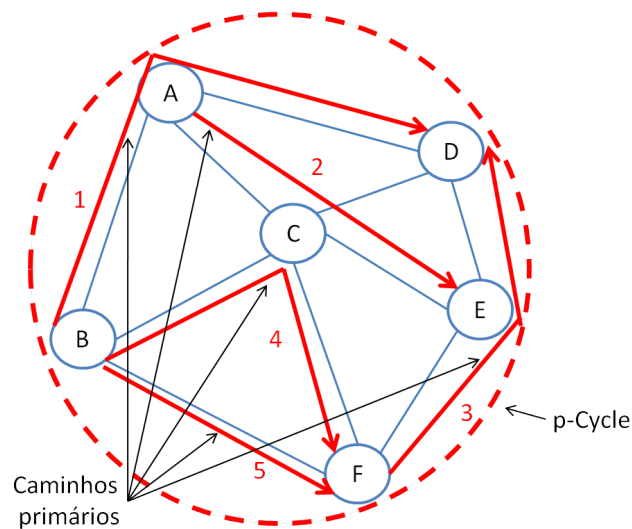


Figura 2.8: Exemplo de *p-cycle* FIPP.

A Figura 2.8 ilustra o conceito de *p-cycle* FIPP. Na Figura 2.8, o grafo representa enlaces bidirecionais, o círculo tracejado ADEFB representa o *p-cycle* e as setas em destaque mostram os vários caminhos que são protegidos por ele. Esses caminhos são tanto caminhos transzonais ao *p-cycle* quanto caminhos no *p-cycle*. O caminho primário 1 é um exemplo de caminho no ciclo, visto que todos os enlaces são utilizados pelo *p-cycle*. O caminho primário 4 representa um caminho transzonal ao *p-cycle*, visto que seus enlaces são disjuntos dos enlaces utilizado no *p-cycle*. Pode-se observar que um único *p-cycle* protege um conjunto de seis caminhos mutuamente excludentes e que possuem seus nós de extremidade no *p-cycle* de caminho.

2.5 Roteamento multicaminho

O roteamento multicaminho (multipath routing) objetiva explorar os recursos da rede física através da utilização de múltiplos caminhos entre os pares origem-destino [96, 11, 9, 10]. O roteamento multicaminho divide a banda passante em vários caminhos, permitindo a rede prover taxas de transferência de dados maiores do que no tradicional roteamento por um único caminho (singlepath routing). O roteamento multicaminho tem sido intensamente investigado em diferentes contextos da literatura. Cidon et al. em [14] realiza uma análise comparativa entre o roteamento de caminho único e roteamento multicaminho, mostrando que este último impacta positivamente no tempo de reserva de alocação de recursos.

Motivado pela necessidade de provisionamento eficiente de conexões e utilização eficiente de recursos, o roteamento multicaminho é utilizado para a agregação de conexões. Padhi et al. [79] investiga o impacto da divisão assimétrica das conexões sobre o problema de agregação de tráfego multicaminho.

O roteamento multicaminho possui diversas vantagens, dentre elas a utilização de banda passante, o balanceamento de carga e a qualidade de serviço (QoS). O uso de multicaminho contribui com a distribuição do tráfego e reduz o atraso fim-a-fim e a perda de pacotes. Entretanto, o atraso de propagação pode aumentar, caso as rotas utilizadas sejam mal escolhidas.

2.6 Agregação de tráfego em redes ópticas

A agregação de tráfego (TG) é uma técnica amplamente adotada para aumentar o desempenho das redes tradicionais multiplexadas por divisão de comprimento de onda (WDM). Agregação de tráfego em redes ópticas elásticas aumenta a eficiência espectral, a utilização de dispositivos de rede e reduz o custo da rede e o consumo de energia. No entanto, o tratamento de agregação de tráfego pode levar à necessidade de mais conversões óptica-elétrica-óptica (O-E-O), bem como processamento de tráfego nos roteadores de núcleo intermediários [22].

Um novo esquema de agregação de tráfego óptica tem sido proposto em EON [115] e [25]. Este esquema agrega requisições de tráfego originados no mesmo nó de origem e, em seguida, separa-os opticamente em nós intermediários usando comprimento de onda seletivo e comutadores de largura de banda variável sem causar interferência. Esses comutadores necessitam de nó intermediário avançado para realizar a função de transmissão e seleção, como consequência da eliminação de conversão óptica-elétrica-óptica. Este esquema de agregação de tráfego óptico combina várias conexões em um caminho óptico sem a necessidade de ter bandas de proteção entre elas [115]. Essa agregação é realizada para suportar agregação de modo transparente, sem conversão do sinal do domínio óptico para elétrico. Esse grupo de caminhos ópticos é chamado de túnel óptico.

A Figura 2.9 ilustra um exemplo de agregação de tráfego óptica em EONs. Na figura, as requisições 1 e 2 são originadas no mesmo nó fonte.

No estudo de sobrevivência em rede ópticas, quando qualquer enlace de fibra no caminho primário falhar, a conexão é redirecionada em um caminho de *backup*. Na forma tradicional de compartilhamento de *backup*, um único caminho de *backup* pode proteger mais de um caminho primário, desde que sejam totalmente disjuntos. A elasticidade dos transponders ópticos oferece uma nova oportunidade de compartilhamento de espectro (sobreposição de espectro).

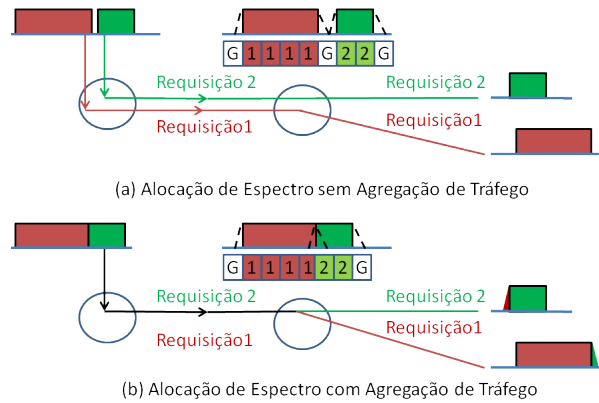


Figura 2.9: Exemplo de agregação de tráfego em EONs.

Dois caminhos de *backup* podem sobrepor o espectro se os caminhos primários das duas conexões são enlaces disjuntos, e se seus caminhos de *backup* percorrerem dois caminhos ópticos adjacentes em um enlace de fibra. O esquema de agregação de tráfego que permite a sobreposição do espectro em redes ópticas elásticas diminui a alocação de recursos pelos caminhos de *backup*, visto que um espectro pode servir a mais de um caminho de *backup* [54].

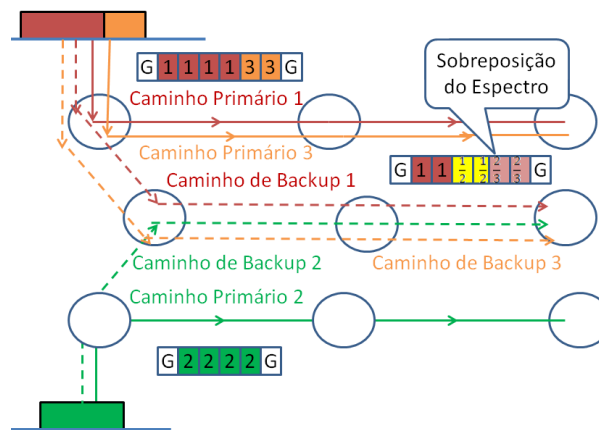


Figura 2.10: Exemplo de sobreposição de espectro EONs.

A Figura 2.10 ilustra um exemplo de sobreposição de espectro. Na figura, as requisições 1, 2 e 3 possuem tanto o caminho primário quanto o de *backup*. Nesta mesma imagem, a técnica de agregação de tráfego é mostrada tanto no caminho primário quanto de *backup*. O caminho primário da requisição 1 e da requisição 3 são disjuntos do caminho primário da requisição 2. Em caso de falha em um enlace, apenas uma das conexões fará uso do espectro sobreposto, tornando possível a sobreposição de espectro entre os caminhos de *backup* destas requisições. A figura mostra dois espectros sobrepostos nos caminhos de *backup* das requisições 1 e 2, e dois espectros se sobrepondo nos caminhos de *backup* das requisições 2 e 3.

2.7 Multigrafo

Quando um grafo possui mais de uma aresta interligando os mesmos dois vértices diz-se que este grafo possui arestas múltiplas (ou arestas paralelas). Ele é chamado de multigrafo ou grafo

múltiplo [39, 35].

Um multigrafo G é uma tripla ordenada $(V(G), E(G), r)$ onde $V(G)$ é o conjunto de vértices do grafo, $E(G)$ é o conjunto de arestas do grafo, e $r : E \rightarrow \{\{x, y\} : x, y \in V\}$ é uma função de $E(G)$ para o conjunto de todos os pares (não ordenados) de dois elementos (não necessariamente distintos) de $V(G)$. Alguns autores também consideram multigrafos aqueles que têm laços [40, 17], isto é, uma aresta que conecta um vértice a ele mesmo; outros chamam estes de pseudografos, reservando o termo multigrafo para os casos em que não há laços. Nesta Tese, grafos com laços são considerados pseudografos.

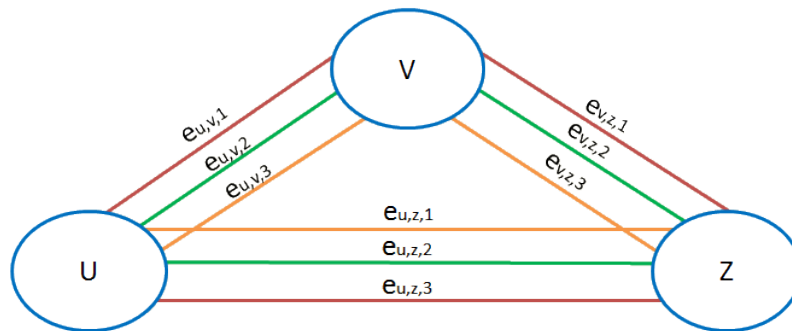


Figura 2.11: Exemplo de multigrafo.

Na Figura 2.11 o grafo possui três arestas entre os vértices U e V, três arestas entre os vértices U e Z e três arestas entre os vértices V e Z, caracterizando-o como um multigrafo.

2.8 Algoritmo genérico

Nesta seção apresenta-se o algoritmo 1, isto é, um método heurístico para resolver de forma genérica o problema de proteção de redes ópticas elásticas com multiplexação espacial. Todos os algoritmos apresentados nesta Tese correspondem a variações e combinações deste meta algoritmo. Para toda nova requisição aceita, o algoritmo escolhe um caminho primário e um caminho de proteção, que satisfaçam a requisição. Para toda requisição deverá ser escolhido ambos os caminhos. No entanto, um caminho de proteção estabelecido para uma requisição pode ser reutilizado por outra requisição, desde que, obedeçam as restrições imposta pelo tipo de proteção de caminho utilizado. Um caminho de proteção somente será buscado ou escolhido se um caminho primário já tiver sido escolhido. Os caminhos correspondem a um conjunto de enlaces, um núcleo e uma faixa de espectro a serem utilizados. Se um caminho primário e/ou caminho de proteção não puderem ser encontrados para a requisição, a requisição é bloqueada.

2.9 Síntese do capítulo

Neste capítulo, foram expostos e analisados os conceitos básicos e o problema de sobrevivência em redes ópticas. Além disso, apresentaram-se conceitos substancialmente relevantes sobre técnicas de proteção, cujo entendimento é fundamental para a compreensão do problema estudado. Introduziram-se definições da técnica de proteção *p-cycle*. Foi, também, descrito o problema de

Algoritmo 1 Algoritmo genérico

Saída: Caminho primário e Caminho de *backup*

- 1: Encontra o menor caminho para estabelecer como caminho primário
 - 2: **Se** Não é possível encontrar caminho primário **então**
 - 3: bloqueia requisição
 - 4: **Senão**
 - 5: Tenta compartilhar um caminho de proteção já estabelecido
 - 6: **Se** Encontrou caminho de proteção para compartilhar **então**
 - 7: Estabelece os caminhos
 - 8: **Senão**
 - 9: **Se** Encontrou caminho para estabelecer como caminho de proteção **então**
 - 10: Estabelece os caminhos
 - 11: **Senão**
 - 12: Bloqueia a requisição
 - 13: **Fim Se**
 - 14: **Fim Se**
 - 15: **Fim Se**
-

roteamento multicaminho. Foi descrito o problema de agregação de tráfego para redes ópticas elásticas. Finalmente, foi apresentado o conceito de multigrafo.

Capítulo 3

Proteção em redes ópticas elásticas com multiplexação espacial

As tecnologias de multiplexação por divisão de comprimento de onda (WDM) suprimiram o crescimento de tráfego de redes na última década. No entanto, o crescimento de tráfego continua exponencial, impulsionado pelo tráfego gerado por data centers e novos serviços emergentes, surgindo, então, a necessidade de implantação de novas técnicas escaláveis e flexíveis. Neste passo, as redes ópticas elásticas com multiplexação espacial sobre fibras multi-núcleo são uma solução promissora para superar o problema de capacidade em redes com aplicações heterogêneas.

Embora algoritmos de Roteamento e alocação de núcleo e espectro tenham sido propostos, apenas [100] estuda proteção de redes ópticas elásticas com multiplexação espacial, isto é um grande problema pois a capacidade de sobrevivência é de suma importância em redes ópticas elásticas com multiplexação espacial, visto que essas carregam enorme quantidade de tráfego. Apesar do acréscimo do número de núcleos aumentar a capacidade da rede, este acréscimo introduz também questões adicionais como *crossstalk* entre núcleos e a complexidade relacionada ao roteamento.

Este capítulo propõe a introdução de três novos algoritmos para proteção em redes ópticas elásticas com multiplexação espacial. Os algoritmos apresentados diferem pelos diferentes tipos de caminhos de *backup* utilizados. Para avaliar a efetividade dos algoritmos propostos, foram realizadas simulações utilizando três topologias derivadas de cenários realistas. Os resultados obtidos mostram que os algoritmos propostos geram menos bloqueio que os outros algoritmos simulados. Além disso, os algoritmos propostos possuem diferentes comportamentos associados à topologia utilizada.

Este capítulo está organizado conforme segue. Na Seção 3.1, apresenta-se um revisão bibliográfica. Na seção 3.2, introduz-se uma técnica para diminuir a complexidade de roteamento dos algoritmos apresentado. Na seção 3.3 apresenta-se a notação utilizada nos algoritmos apresentados nesse capítulo. Nas Seções 3.4, 3.5 e 3.6 descrevem-se, respectivamente, o algoritmo *Shared Backup Path Protection for MultiCore network* (SBPPMC), o algoritmo *Failure-Independent Path Protecting for MultiCore network* (FIPPMC) e o algoritmo *Minimum Interference and Failure-independent path protecting for MultiCore networks* (MIFMC). Na Seção 3.7, compara-se o desempenho dos algoritmos propostos e de outros dois algoritmos. Finalmente, na Seção 3.8, apresenta-se a síntese do capítulo.

3.1 Trabalhos relacionados

Na literatura, diversos trabalhos estudam proteção em redes ópticas, no entanto, dentre todos os trabalhos encontrados, apenas o [100] trata de proteção de redes ópticas elásticas utilizando multiplexação espacial.

Os autores em [33] analisam as vantagens da utilização da arquitetura de rede óptica elástica, dando enfoque em tráfego para nuvens. Eles realizam uma avaliação no impacto dos desastres sobre o desempenho de redes ópticas elásticas. Comparam, também, duas abordagens de proteção, proteção de caminho dedicado (DPP) e proteção de caminho de *backup* compartilhada (SBPP), utilizando métricas tais como: consumo de energia e utilização do espectro.

Os autores de [93] propuseram políticas conservadoras de proteção em redes ópticas baseadas em OFDM com alocação de banda elástica. Introduziram uma política em que caminhos ópticos de *backup* protegem caminhos ópticos primários, utilizando caminhos disjuntos, o que aumenta a robustez à proteção.

Foram propostos em [69] dois novos algoritmos para fornecer proteção de caminho através do emprego de *p-cycle* de caminho em redes ópticas elásticas. Os algoritmos proporcionam proteção contra uma falha, bem como proteção contra duas falhas simultâneas.

Os autores em [110] enfocam a proteção de caminho de *backup* compartilhado com base na conectividade de conteúdo contra desastres no contexto de redes ópticas elásticas de datacenter. O artigo propõe uma formulação de programação linear inteira para minimizar tanto a capacidade de reposição quanto os recursos de trabalho em um data center. Além disso, propõe um algoritmo heurístico para grandes redes de data centers.

Em [68], é proposto um algoritmo para prover proteção de caminho independente de falhas utilizando *p-cycle* para redes ópticas elásticas. O algoritmo foi comparado com dois algoritmos encontrados na literatura.

Os autores em [58] propõem um algoritmo de RSCA baseado no algoritmo *Connected Component Labelling* (CCL). Nele, o algoritmo RSCA representa o espectro de fibras multi-núcleos como matrizes e o algoritmo CCL descobre o espectro disponível para alocar uma solicitação de conexão. Políticas de ajuste do espectro também são propostas para serem usadas em conjunto com o algoritmo CCL.

Em [102], o problema de RSCA é dividido em problema de roteamento e SCA, e introduz um método de pré-computação de *k*-caminhos como uma solução de roteamento.

Os autores em [62] desenvolvem uma formulação de programação linear inteira (ILP) para o problema de planejamento de rede RSCA, priorizando a eficiência espectral nos núcleos da rede óptica elástica com multiplexação espacial. Eles utilizam sete núcleos por fibra em suas simulações.

Tode *et al.* [103] propõe métodos RSCA em redes ópticas elásticas com multiplexação espacial. O trabalho introduz conceitos de área priorizada e dois tipos de ciência de *crosstalk*.

Em [100], foi proposto o algoritmo Cap-DPP que utiliza caminhos dedicados para realizar proteção, e utiliza uma técnica de *k*-caminhos para encontrar o caminho primário e o caminho de *backup* em redes ópticas elásticas com multiplexação espacial.

A Tabela 3.1 apresenta uma comparação entre os trabalhos relacionados e os algoritmos propostos neste capítulo, os trabalhos foram categorizados com relação à proteção, ao uso de multiplexação de divisão espacial, ao uso de roteamento e alocação de espectro, ao uso de

roteamento e alocação de núcleo e espectro, e ao uso do *p-cycle* para fornecer proteção e, por fim, ao uso de proteção utilizando caminhos de *backup* compartilhados.

Tabela 3.1: Comparação dos trabalhos relacionados com os algoritmos FIPPMC, SBPPMC, MIFMC.

Abordagem	Proteção	SDM	RSA	RSCA	<i>p-Cycle</i>	SBPP
Goscien <i>et al.</i> [33]	Sim	Não	Sim	Não	Não	Sim
Sillard <i>et al.</i> [93]	Sim	Não	Sim	Não	Não	Sim
Oliveira <i>et al.</i> [69]	Sim	Não	Sim	Não	Sim	Não
Xie <i>et al.</i> [110]	Sim	Não	Sim	Não	Não	Sim
Oliveira <i>et al.</i> [68]	Sim	Não	Sim	Não	Sim	Não
Moura <i>et al.</i> [58]	Não	Sim	Não	Sim	Não	Não
Tode <i>et al.</i> [102]	Não	Sim	Não	Sim	Não	Não
Muhammad <i>et al.</i> [62]	Não	Sim	Não	Sim	Não	Não
Tode <i>et al.</i> [103]	Não	Sim	Não	Sim	Não	Não
Yuanlong <i>et al.</i> [100]	Sim	Sim	Não	Sim	Não	Não
FIPPMC	Sim	Sim	Não	Sim	Sim	Não
SBPPMC	Sim	Sim	Não	Sim	Não	Sim
MIFMC	Sim	Sim	Não	Sim	Sim	Não

3.2 Multigrafo

A adição da dimensão espacial na multiplexação aumenta a complexidade da atribuição do espectro, uma vez que o espectro de todos os núcleos tem de ser levado em consideração durante a atribuição.

A fim de utilizar baixa complexidade e levar em consideração características de *slots* e núcleos em conjunto com o roteamento e proteção dos algoritmos propostos neste capítulo, os algoritmos modelam a disponibilidade do espectro na rede como um multigrafo (Figura 3.1(a)). No multigrafo, um rótulo em uma aresta representa a disponibilidade de um *slot*. Na figura 3.1(b), o multigrafo é dividido em C multigrafos, onde C é o número de núcleos. Cada aresta nos multigrafos criados representa um slot em um determinado núcleo. Cada um desses multigrafos da Figura 3.1(b) é transformado em um multigrafo com $N - b + 1$ arestas (Figuras 3.1(c) e 3.1(d)), onde b é a demanda de largura de banda da requisição em *slot* e N o número de *slots* em cada núcleo. Cada aresta do multigrafo da Figura 3.1(d) representa b arestas contínuas do multigrafo da Figura 3.1(c). Essa representação garante a contiguidade do espectro na solução. Por fim, cada aresta de cada multigrafo resultante formará um grafo (Figura 3.1(e)). Os grafos gerados serão utilizados para encontrar os caminhos. Em outras palavras, o multigrafo original é transformado em $C \times (N - b + 1)$ grafos. Cada aresta nestes grafos representa uma combinação de b *slots* em um único núcleo. Esta representação assegura a contiguidade do espectro à solução, uma vez que os grafos gerados são formados por apenas arestas contíguas. Nestes grafos, um valor ∞ significa que pelo menos um *slot* já está alocado enquanto que um valor menor significa que todos os *slots* estão disponíveis para alocação. O valor ∞ pode significar também que o slot não pode ser utilizado, visto que a utilização do mesmo gerará *crossstalk* inaceitável para a transmissão atual ou para uma transmissão que utiliza um slot no núcleo vizinho. Em

outras palavras, um slot não estará disponível se já estiver alocado ou se o valor de *crosstalk* no slot for inaceitável para uma transmissão bem-sucedida, ou ainda se a utilização deste slot tornar uma transmissão que utiliza um slot do núcleo adjacente inaceitável.

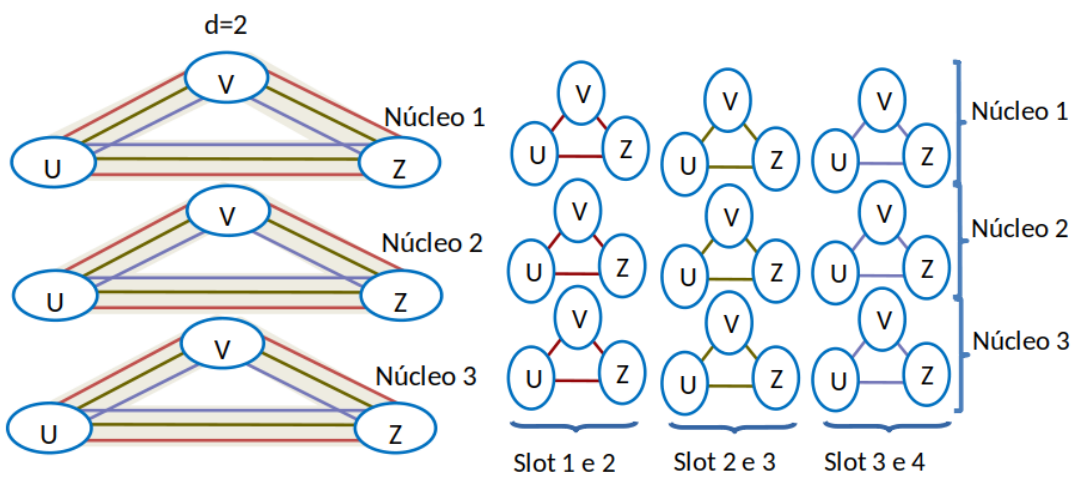
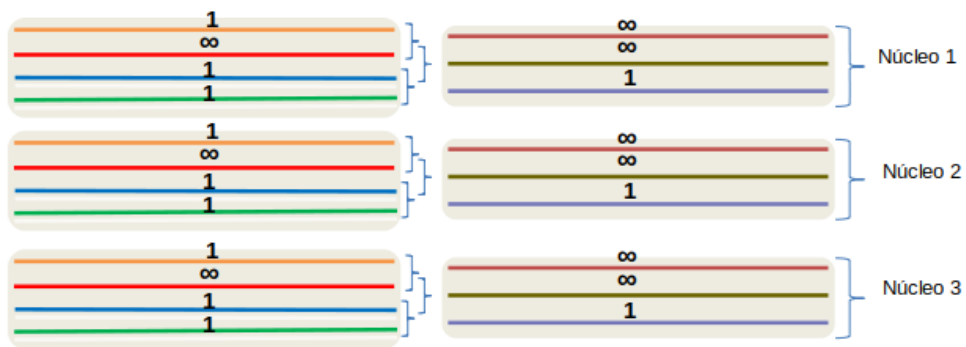
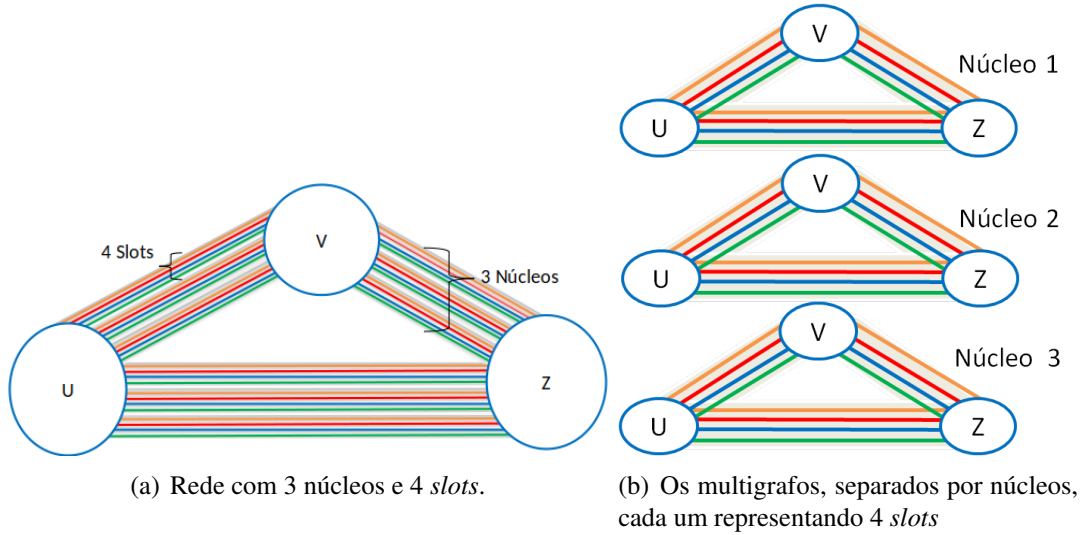


Figura 3.1: Transformando multigrafo em grafos.

3.3 Notação

A seguinte notação matemática será usada neste capítulo:

- s : nó fonte;
- d : nó destino;
- b : demanda de largura de banda mais banda de guarda;
- F : número de enlaces físicos;
- N : número do conjunto de *slots* entre dois nós;
- C : número de núcleos entre dois nós;
- V : conjunto de nós;
- $e_{u,v,n}$: A n -ésima aresta conectando u e v . Cada aresta representa um *slot* entre os nós u e v .
- $E = \{e_{u,v,n}\}$: conjunto de arestas;
- $G = (V, E, W)$: multigrafo composto por um conjunto de nós V , um conjunto de arestas E e um conjunto de pesos de arestas W , $|E| = C \cdot N \cdot F$;
- $r(s, d, b)$: requisição do nó s para o nó d com demanda b ;
- $w(e_{u,v,n})$: peso da aresta $e_{u,v,n}$; $w(e_{u,v,n}) < \infty$ se o n -ésimo *slot* no enlace conectando os nós u e v estão livres e $w(e_{u,v,n}) = \infty$ se o *slot* já estiver alocado;
- $W = \{w(e_{u,v,n})\}$: conjunto de pesos das arestas;
- $\tau(G, C, b) = \{\tilde{G}_{n,b}\}$: função que produz todos os σ grafos de G ;
- $\sigma = |\{\tilde{G}_{n,b}\}| = C \times (N - b + 1)$: número de grafos extraído do multigrafo;
- $\tilde{V} = V$: conjunto de nós dos grafos gerados;
- $\tilde{W} = \{\tilde{w}_n(\tilde{e}_{\tilde{u},\tilde{v}})\}$: conjunto de pesos das arestas dos grafos gerados;
- $\tilde{G}_{n,b} = (\tilde{V}, \tilde{E}, \tilde{W})$: o n -ésimo grafo tal que \tilde{E} é o conjunto de arestas conectando $\{\tilde{u}, \tilde{v}\} \in \tilde{V}$ e \tilde{W} é o conjunto de custos associados a \tilde{E} . A aresta \tilde{E} corresponde ao mapeamento de b aresta em G iniciando na n -ésima aresta;
- $\tilde{e}_{\tilde{u},\tilde{v}} = \{e_{u,v,n}\} \in \tilde{E}$ é uma sequência tal que $e_{u,v,n}$ é a menor aresta ordenada, $e_{u,v,n+b}$ é a maior aresta ordenada $|\tilde{e}_{\tilde{u},\tilde{v}}| = b$;
- $\tilde{w}_n(\tilde{e}_{\tilde{u},\tilde{v}})$: peso da aresta $\tilde{e}_{\tilde{u},\tilde{v}}$ no grafo n ; $\tilde{w}_n(\tilde{e}_{\tilde{u},\tilde{v}}) < \infty$ se as arestas correspondes no grafo estão livres e podem ser utilizadas sem gerar XT inaceitável e $\tilde{w}_n(\tilde{e}_{\tilde{u},\tilde{v}}) = \infty$ se aresta já estiver alocado ou se gerar XT inaceitável;
- $\delta(\tilde{G}_{n,b}, r(s, d, b))$: função que produz menor caminho entre s e d para o grafo $\tilde{G}_{n,b}$, que satisfaz a requisição de b *slots*;
- P_n : sequência de \tilde{G}_n tal que o nó fonte s é o menor nó ordenado e d é o maior nó ordenado;
- $W(P_n)$: o peso do caminho P_n , a soma dos pesos de todas as arestas na sequência;
- $W_{P_{s,d}}$ = peso do menor caminho entre s e d ;
- Z_n : caminho de *backup* contendo os vértices u e v e arestas correspondendo ao mapeamento de b arestas do multigrafo G ;
- Z : conjunto de todos os caminhos de *backup* ativos;
- P_{Z_n} : conjunto de todos os caminhos protegidos pelo caminho de *backup* Z_n ;
- $W(Z_n)$: o peso do caminho de *backup* Z_n (a soma dos pesos de todas as arestas da sequência);
- $W_{Z_{s,d}}$ = peso do menor caminho de *backup* entre s e d ;
- $\nu(P_n, Z, r(s, d, b))$: função que procura um caminho de *backup* Z_n em Z em que P_{Z_n} são caminhos disjuntos de P_n e satisfazem a requisição de banda passante b e que s e d são nós

inicias e finais do caminho;

$\beta(\tilde{G}_{n,b}, P_n, r(s, d, b))$: função que produz um menor caminho de *backup* Z_n entre s and d em $\tilde{G}_{n,b}$, em que Z_n e P_{Z_n} são caminhos disjuntos para P_n , e que o caminho satisfaz a requisição b ;

T_n : p -cycle contendo os vértices u e v e arestas correspondendo ao mapeamento de b arestas do multigrafo G ;

T : conjunto de todos os p -cycles ativos;

P_{T_n} : conjunto de todos os caminhos protegidos pelo p -cycle T_n ;

$W(T_n)$: o peso do p -cycle T_n (a soma dos pesos de todas as arestas);

$W_{T_{s,d}}$ = peso do p -cycle que protegerá o caminho entre s e d ;

$\mu(P_n, T, r(s, d, b))$: função que procura um p -cycle T_n em T que P_{T_n} são caminhos disjuntos de P_n e satisfazem a requisição b e que s e d fazem parte do p -cycle;

$\xi(\tilde{G}_{n,b}, P_n, r(s, d, b))$: função que produz um menor p -cycle T_n , entre s and d em $\tilde{G}_{n,b}$, que P_{T_n} são caminhos disjuntos para P_n , e que o p -cycle satisfaz a requisição b ;

$\vartheta(\tilde{G}_{n,b}, P_n, r(s, d, b))$: função que produz um p -cycle T_n , entre s and d em $\tilde{G}_{n,b}$, que satisfaz a requisição b , é disjunto de P_n (p -cycle é transzonal a P_n) e que P_{T_n} são caminhos disjuntos para P_n ;

3.4 O algoritmo SBPPMC

O algoritmo introduzido, nesta seção, chamado *Shared Backup Path Protection for MultiCore network* (SBPPMC), decide sobre o estabelecimento de conexões em uma rede com proteção contra uma falha. Nele, um caminho óptico é estabelecido se e somente se ele pode ser protegido por um caminho de *backup* compartilhado (SBPP).

O caminho de *backup* compartilhado é uma técnica de proteção que tem sido amplamente investigada nos últimos anos devido ao compartilhamento de recursos eficiente para realizar a proteção, bem como a flexibilidade no fornecimento de serviços. A técnica SBPP emprega um esquema de proteção 1:N no qual um único caminho de *backup* pode proteger mais de um caminho primário, desde que seus caminhos primários correspondentes sejam disjuntos.

No algoritmo SBPPMC, a linha 1 transforma o multigrafo em $C \times (N - b + 1)$ grafos. A linha 2 resolve um algoritmo de menor caminho para os grafos gerados utilizando Dijkstra. A linha 3 escolhe o menor caminho e o peso do caminho dentre os menores caminhos dos grafos. Na linha 4, as arestas correspondentes ao caminho têm os pesos configurados para ∞ , significando que os *slots* foram alocados. Se o peso do menor caminho é ∞ , não foi possível encontrar um caminho sob a restrição de contiguidade para a demanda b (linha 5). Então, a requisição será bloqueada na (linha 6). Caso contrário, um caminho de *backup* para proteger o caminho selecionado será buscado a fim de ser compartilhado dentre os caminhos de *backup* ativos (Linha 8). Caso exista um caminho de *backup* que possa proteger o caminho, o caminho é estabelecido (linha 9). Caso não haja um caminho de *backup*, ele necessita ser criado (linha 11). A criação do caminho de *backup* considera que todos os enlaces utilizados no caminho primário P_n terão peso ∞ , isso garante que os dois caminhos sejam disjuntos. Além disso a criação do caminho considera o algoritmo de Dijkstra e o grafos gerados do multigrafo para obtenção do menor caminho válido. Caso não se possa criar um caminho de *backup* (linha 13),

Algoritmo 2 SBPPMC

Entrada: $G, r(s, d, b)$
Saída: Caminho primário e Caminho de *backup*

```

1:  $\tau(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\tilde{G}_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} = \infty$  então
6:   bloqueia requisição  $(r(s, d, b))$ 
7: Senão
8:   Se  $\exists \nu(P_n, Z, r(s, d, b))$  então
9:     estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
10:  Senão
11:     $(W(Z_n), Z_n) = \beta(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
12:     $W_{Z_{s,d}} = W(Z_n) \mid \forall i W(Z_n) \leq W(Z_i)$ 
13:    Se  $W_{Z_{s,d}} = \infty$  então
14:      bloqueia requisição  $(r(s, d, b))$ 
15:    Senão
16:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
17:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in Z_i$ 
18:    Fim Se
19:  Fim Se
20: Fim Se

```

a requisição é bloqueada (linha 14). Do contrário, o caminho de *backup*, será estabelecido e as arestas correspondentes terão seus pesos configurados para ∞ (linhas 17).

A complexidade do algoritmo SBPPMC é analisada como: a complexidade de transformar o multigrafo em grafos é $O(E+V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para o caminho de *backup*, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos. Logo, no pior caso, o algoritmo SBPPMC tem complexidade de $(E+V) + 2 \times (C \times (N - b) \times (E + V \log V)) + E + V$. Como N, C e b podem ser expressos como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

3.5 O algoritmo FIPPMC

A utilização de técnicas de compartilhamento de caminhos atribui para cada caminho primário um caminho de *backup* pré-calculado. Com isso, um conjunto de caminhos primários podem compartilhar um mesmo caminho de *backup*, se os seus caminhos primários são mutuamente disjuntos. Nos algoritmos que utilizam essa técnica, quando ocorre uma falha, os nós ao longo de uma rota de *backup* devem ser reconfigurados para utilizar o caminho de *backup* correspondente ao caminho de trabalho que falhou. Isso gera atrasos na sinalização e configuração dos nós intermediários, aumentando assim o tempo de restauração. Um *p-cycle* é um caminho fechado pré-conectado e cíclico de capacidade reservada para realizar proteção. Um caso especial de *p-cycle* é o *p-cycle* FIPP, que possui tempo de comutação de proteção tipicamente na ordem de 50 ms [82] e 10 ms [82] para detecção de falha, o que o torna uma técnica de proteção

mais rápida que mecanismos comuns, como caminhos de *backup* compartilhados que possuem tempo de comutação de proteção de aproximadamente 200 ms [6].

O algoritmo proposto nesta seção, chamado *Failure-Independent Path Protecting for MultiCore network* (FIPPMC), decide sobre o estabelecimento de caminhos ópticos em uma rede protegida por *p-cycles* FIPP. Nele, um caminho óptico é estabelecido se e somente se, ele pode ser protegido por um *p-cycle* FIPP, que pode ter tanto caminhos transzonais como caminhos no ciclo. Um caminho transzonal a um *p-cycle* refere-se aos caminhos que possuem os nós finais no *p-cycle*, mas pelo menos um nó intermediário não estará no *p-cycle*. O algoritmo FIPPMC, possui a mesma estrutura do algoritmo SBPPMC. Primeiramente, o algoritmo gera grafos a partir do multigrafo, dentre esses grafos busca-se o menor caminho que será o caminho primário. Posteriormente, busca-se um caminho de *backup* pré-existente que possa ser compartilhado, caso não exista gera-se um caminho para proteção. No algoritmo FIPPMC, o caminho de *backup* é um *p-cycle* FIPP.

Algoritmo 3 FIPPMC

Entrada: $G, r(s, d, b)$

Saída: Caminho primário e Caminho de *backup*

- 1: $\tau(G, C, b)$
 - 2: $(W(P_n), P_n) = \delta(\tilde{G}_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$
 - 3: $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$
 - 4: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$
 - 5: **Se** $W_{P_{s,d}} = \infty$ **então**
 - 6: bloqueia requisição ($r(s, d, b)$)
 - 7: **Senão**
 - 8: **Se** $\exists \mu(P_n, T, r(s, d, b))$ **então**
 - 9: estabelece requisição ($r(s, d, b)$) **como** P_n **e** T_n
 - 10: **Senão**
 - 11: $(W(T_n), T_n) = \xi(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$
 - 12: $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$
 - 13: **Se** $W_{T_{s,d}} = \infty$ **então**
 - 14: bloqueia requisição ($r(s, d, b)$)
 - 15: **Senão**
 - 16: estabelece requisição ($r(s, d, b)$) **como** P_n **e** T_n
 - 17: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$
 - 18: **Fim Se**
 - 19: **Fim Se**
 - 20: **Fim Se**
-

O processo de geração do caminho primário é idêntico tanto para o SBPPMC quanto para o FIPPMC, isso pode ser observado nas linhas de 1 a 6. No entanto, para o algoritmo FIPPMC, um *p-cycle* para proteger o caminho criado nas linhas de 1 a 6 será buscado dentre os *p-cycles* ativos (linha 8). Os *p-cycles* ativos são aqueles que já foram criados e protegem uma conexão na rede. Caso exista um *p-cycle* que possa proteger o caminho, o caminho é estabelecido (linha 9). Caso não haja um *p-cycle* ativo que possa proteger o caminho, um *p-cycle* necessita ser criado (linha 11). Para criação do *p-cycle*, pressupõe-se que um *p-cycle* pode ser visto como a combinação de dois caminhos disjuntos entre dois nós finais. Com isso, utilizou-se o algoritmo de Suurballe [5] para encontrar os dois caminhos disjuntos mais curtos, então um ciclo é formado. Caso não possa criar um *p-cycle* (linha 13), a requisição é bloqueada. Caso contrário, o *p-cycle*

será estabelecido e as arestas correspondentes têm os pesos configurados para ∞ (Linhas 17). É importante notar que o algoritmo FIPPMC executa algoritmo de Suurballe em cada grafo gerado, de forma que obrigatoriamente as restrições de continuidade de núcleo e espectro serão mantidas na formação do p -cycle pelos dois caminhos encontrados.

A complexidade do algoritmo FIPPMC é analisada como: a complexidade de transformar o multigrafo em grafos é $O(E+V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, onde a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para encontrar o p -cycle, o algoritmo de Suurballe é executado em $C \times (N - b)$ grafos, onde a complexidade de Suurballe é $O(E + V \log V)$ [5]. Logo, no pior caso, o algoritmo FIPPMC tem complexidade de $(E + V) + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V)$. Como N , C e b podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

3.6 O algoritmo MIFMC

Os tradicionais algoritmos de proteção levam a uma rápida saturação de enlaces da rede, impulsionando novos algoritmos, especialmente aqueles que empregam interferência mínima para promover uma utilização equilibrada dos recursos. Algoritmos de interferência mínima geram caminhos com menor interferência entre os caminhos de requisições distintas [26, 27]. Um p -cycle protegendo uma requisição pode sobrecarregar enlaces, uma vez que o p -cycle pode usar os mesmos enlaces do caminho primário. A abordagem desta seção, sempre que possível, gera p -cycle transzonal ao caminho primário, evitando que o p -cycle e o caminho primário de uma mesma requisição utilizem o mesmo enlace, evitando, também, a saturação de enlaces da rede, e portanto, minimizando a rejeição de requisições futuras.

Como o algoritmo FIPPMC, o algoritmo introduzido nesta seção, chamado *Minimum Interference and Failure-independent path protecting for MultiCore networks* (MIFMC), também, decide sobre provisão de proteção dos caminhos ópticos através do uso de p -cycles FIPP. No entanto, diferentemente do algoritmo FIPPMC, o algoritmo MIFMC prioriza o uso de p -cycles transzonais ao caminho primário, gerando interferência mínima e reduzindo rejeições de requisições futuras. O algoritmo 4 garante um caminho de *backup* para cada caminho óptico estabelecido e a proteção é garantida para falhas únicas. A reserva de recursos para criar o p -cycle FIPP, protegendo uma requisição, pode sobrecarregar os enlaces na rede, uma vez que o p -cycle que protege os caminhos no p -cycle usam os mesmos enlaces do caminho primário. Por outro lado, o p -cycle que protege caminhos transzonais tende a reservar mais recursos, pois tem um maior número de saltos. Os p -cycles que protegem caminhos no p -cycle utilizam menos recursos da rede do que p -cycle que protegem caminhos transzonais, mas, por outro lado, podem sobrecarregar os enlaces ao longo de um caminho. Por conseguinte, é necessário adotar critérios para evitar a formação de gargalos, equilibrando a carga entre potenciais caminhos, sendo importante também adotar uma abordagem de interferência mínima para evitar o bloqueio de requisições.

No algoritmo MIFMC, a criação do caminho primário é similar aos algoritmos FIPPMC e SBPPMC, como pode ser visto pelas linhas de 1 a 6. A linhas 8 a 10 do algoritmo MIFMC são semelhantes ao algoritmo FIPPMC, pois ambos algoritmos tentam compartilhar os p -cycles ati-

Algoritmo 4 MIFMC

Entrada: $G, r(s, d, b)$
Saída: Caminho primário e Caminho de *backup*

```

1:  $\tau(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\tilde{G}_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) | \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} = \infty$  então
6:   bloqueia requisição( $r(s, d, b)$ )
7: Senão
8:   Se  $\exists \mu(P_n, T, r(s, d, b))$  então
9:     estabelece requisição ( $r(s, d, b)$ ) como  $P_n$  e  $T_n$ 
10:  Senão
11:     $(W(T_n), T_n) = \vartheta(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
12:     $W_{T_{s,d}} = W(T_n) | \forall i W(T_n) \leq W(T_i)$ 
13:    Se  $W_{T_{s,d}} = \infty$  então
14:       $(W(T_n), T_n) = \xi(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
15:       $W_{T_{s,d}} = W(T_n) | \forall i W(T_n) \leq W(T_i)$ 
16:      Se  $W_{T_{s,d}} = \infty$  então
17:        bloqueia requisição ( $r(s, d, b)$ )
18:      Fim Se
19:    Fim Se
20:    Se  $W_{T_{s,d}} \neq \infty$  então
21:      estabelece requisição ( $r(s, d, b)$ ) como  $P_n$  e  $T_n$ 
22:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
23:    Fim Se
24:  Fim Se
25: Fim Se

```

vos de forma a proteger as novas requisições. No entanto, o processo de criação de *p-cycle* no algoritmo MIFMC se dá por duas etapas. Primeiramente, diferentemente do algoritmo FIPPMC, o algoritmo MIFMC prioriza criar *p-cycles* transzonais nos grafos gerados (linha 11), isto é, criar *p-cycles* que não utilizam os mesmos enlaces do caminho primário, utilizando o conceito de interferência mínima.

Neste algoritmo, a criação do *p-cycle* considera os enlaces utilizados no caminho primário marcados como ∞ , garantindo o caminho transzonal ao *p-cycle*. A linha 12 escolhe o menor *p-cycle* transzonal e o peso dentre os menores *p-cycles* dentre os grafos. Caso não seja possível criar um *p-cycle* com interferência mínima, ou seja, um *p-cycle* transzonal ao caminho primário (linha 13), um menor *p-cycle* necessita ser criado (linha 14). A segunda etapa gera o *p-cycle* sem interferência mínima, ou seja, o *p-cycle* pode ser criado utilizando os enlaces do caminho primário (linha 14). Caso não possa criar nenhum dos *p-cycles* (linha 16), a requisição é bloqueada. Caso contrário, o *p-cycle* será estabelecido e as arestas correspondente tem os pesos configurados para ∞ (linhas 21 à 22).

A complexidade do algoritmo MIFMC é analisada como: a complexidade de transformar o multigrafo em grafos é $O(E+V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, onde a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para o caminho de *backup*, o pior caso pode executar o

algoritmo de Suurballe [5] em $C \times (N - b)$ grafos para o p -cycle transzonal e $C \times (N - b)$ grafos para o p -cycle de menor caminho, sendo a complexidade de Suurballe $O(E + V \log V)$. Logo no pior caso, o algoritmo MIFMC tem complexidade de $(E + V) + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V)$. Como N , C e b podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

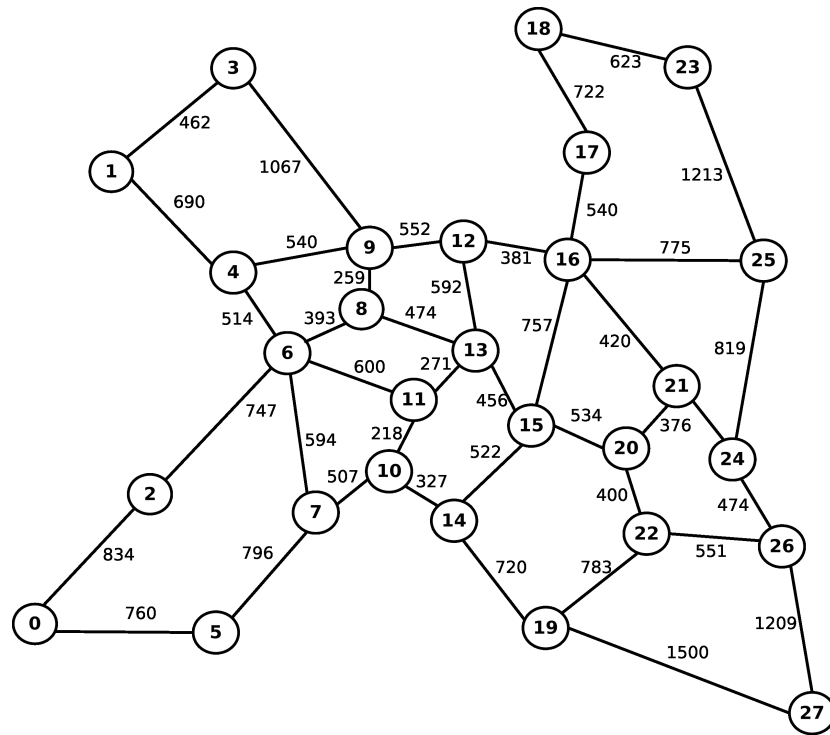
3.7 Resultados numéricos

Para avaliar o desempenho dos algoritmos proposto neste capítulo, experimentos de simulação foram realizados utilizando o simulador FlexGridSim [59]. Em cada simulação, 100.000 requisições foram geradas. Utilizou-se o método de replicação independente e adotou-se nível de confiança de 95% para os intervalos de confiança. Diferentes topologias com diferentes conectividades foram empregadas nas simulações: a Pan-European (Figura 3.2(a)) com 28 nós e 39 enlaces bidirecionais; a USA (Figura 3.2(b)) com 24 nós e 43 enlaces bidirecionais; e a NSF (Figura 3.2(c)) com 14 nós e 18 enlaces bidirecionais. As figuras mostram as respectivas distancias entre os nós. O espectro foi dividido em 320 *slots* de largura de banda de 12.5 GHz cada [8, 87] e as fibras simuladas possuem 7 núcleos [102]. Variou-se a carga entre 25 e 500 erlangs em intervalos de 25 erlangs. Sete tipos de requisições foram utilizadas, as demandas de banda de 25 Gbps, 50 Gbps, 125 Gbps, 200 Gbps, 500 Gbps, 750 Gbps e 1 Tbps. Nas simulações, utilizou-se um limite de *crosstalk* de -16 dB considerado como aceitável [91]. Além disso, utilizou-se o formato de modulação fixo BPSK que realiza a transmissão de 1 bit por símbolo.

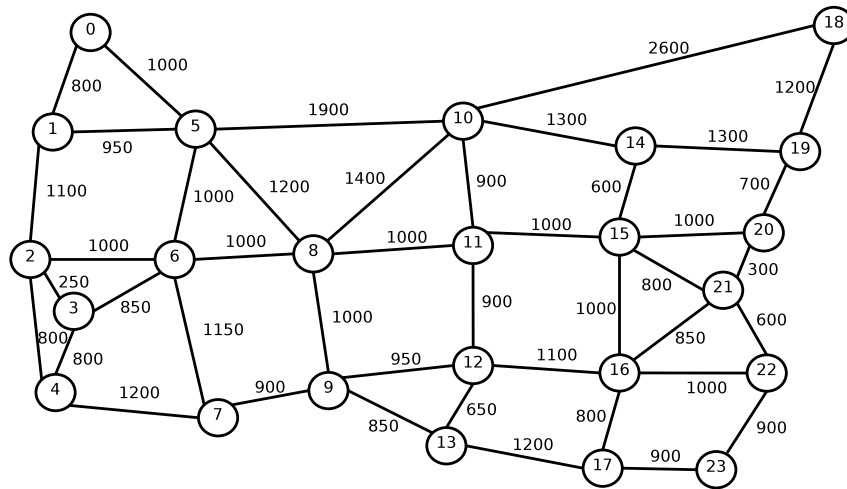
As requisições de conexão são uniformemente distribuídas entre todos os pares de nós da rede. O tempo de espera entre requisições segue uma distribuição exponencial negativa com uma média de uma unidade. A carga da rede, dada em Erlang, é definida como: $A = R \times h \times (B/\lambda)$, onde R é a taxa de chegada da requisição, h é o tempo entre chegadas das requisições, e B é a requisição de largura de banda normalizada para a capacidade do enlace.

Nas figuras, as curvas rotuladas como FIPPMC mostram os resultados para as redes que utilizam o algoritmo FIPPMC (Algoritmo 3), as curvas rotuladas como MIFMC mostram os resultados para as redes que utilizam o algoritmo MIFMC (Algoritmo 4), e as curvas rotuladas como SBPPMC mostram os resultados para as redes que utilizam o algoritmo SBPPMC (Algoritmo 2), as curvas rotuladas como SSCA encontram o caminho primário através do algoritmo proposto em [102], utilizando $k = 3$ e o caminho de *backup* adicionando a capacidade de compartilhamento ao mesmo algoritmo, enquanto as curvas rotuladas como Cap-DPP mostram os resultados para as redes que utilizam o algoritmo proposto em [100], o algoritmo Ca-DPP encontra k caminhos disjuntos, com $k = 2$, o primeiro caminho equivale ao caminho primário e o segundo caminho equivale ao caminho de *backup*.

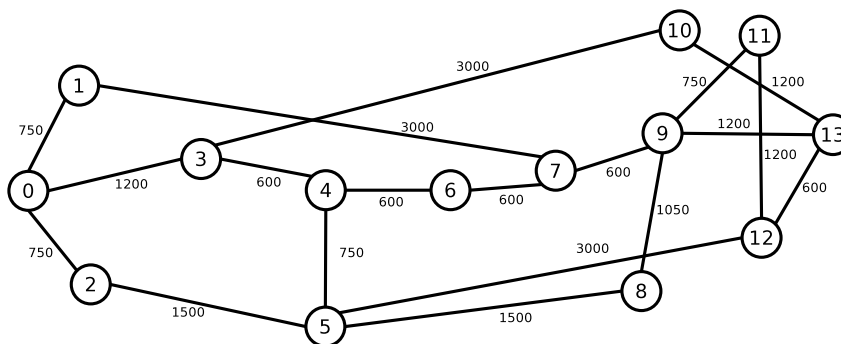
As métricas consideradas para avaliação dos algoritmos deste capítulo são a probabilidade de bloqueio de banda (*Bandwidth Blocking Ratio* - BBR), o valor de *crosstalk* por *slot* (*crosstalk per slot* - CpS), a razão de fragmentação (*Fragmentation Ratio* - FR), a eficiência energética (*Energy Efficiency* - EE), o número de saltos para o caminho primário e o número de saltos para o caminho de *backup*. A BBR é definida pela porcentagem de banda (tráfego) bloqueada sobre o total de banda solicitada durante todo o período de simulação. Conforme Equação 3.1.



(a) Pan-European



(b) USA



(c) NSF

Figura 3.2: Topologias.

$$BBR = \frac{\sum BandaBloqueada}{\sum BandaTotal} \quad (3.1)$$

O valor de CpS é definido como a relação média entre os pares de slots de frequência usados que têm a mesma frequência e estão localizados em núcleos adjacentes (*Arrangement of Crosstalk, AoC*) e o total de *slots* usados [30], como denotado na Equação 3.2.

$$CpS = \frac{AoC}{TotalSlots} \quad (3.2)$$

A FR é gerada a partir do estabelecimento e a desconexão de caminhos. A razão de fragmentação compara o número máximo de *slots* contíguos disponíveis ao número de *slots* disponíveis no enlace [57]. A razão de fragmentação é denotada pela Equação 3.3.

$$FR = \frac{TotalLivre - maiorBlocoLivre}{TotalLivre} \quad (3.3)$$

O número de saltos do caminho primário apresenta a média do número de saltos de todos caminhos primário estabelecidos. O número de saltos do caminho de *backup* apresenta a média do número de saltos de todos caminhos de *backup* estabelecidos. A Equação 3.4 foi utilizada para encontrar o número de saltos médio.

$$MediaSaltos = \frac{\sum SaltosTodosCaminho}{TotalCaminhos} \quad (3.4)$$

A medida de eficiência energética (EE) é obtida dividindo-se a demanda de tráfego total atendida com sucesso na rede (*Total Traffic Network, TTN*) pelo consumo total de energia da rede (*Total Power Consumption, TPC*). O TTN é calculada somando-se a taxa de transmissão total de todas as requisições que foram alocadas com êxito na rede. O TPC é o consumo total de energia dos transponders, dos EDFAs e dos os OXCs na rede, os EDFAs estão distribuídos conforme o tamanho dos enlaces na topologia real.

$$EE = \frac{TTN}{TPC} \quad (3.5)$$

O calculo da eficiência energética foi realizado como em Viscaíno *et.al.*[104], no entanto, os valores da energia consumida pelo OXC foram baseados em Fujii *et. al.* [29], para representar o consumo de uma OXC em redes ópticas elásticas com multiplexação espacial.

A Figura 3.3 mostra a probabilidade de bloqueio para a topologia Pan-European. Enquanto os algoritmos FIPPMC, SBPPMC e MIFMC iniciam o bloqueio de requisições sob cargas de 75 erlangs, o Cap-DPP e o SSCA iniciam o bloqueio de requisições sob cargas de 50 erlangs. Os valores de BBR produzidos pelo MIFMC são sempre mais baixos do que outros algoritmos, isso evidencia o benefício de considerar os critérios de interferência mínima para criar os *p-cycles* durante a escolha da rota de *backup* para topologia com alta conectividade. Os algoritmos FIPPMC e MIFMC produzem comportamento de BBR semelhante, devido à alta conectividade da topologia Pan-European, estes algoritmos que utilizam *p-cycle* não produzem bloqueio até cargas de 75 erlangs. O algoritmo SBPPMC produz maiores valores de BBR que os produzidos pelos algoritmos FIPPMC e MIFMC. Isto ocorre pois para esta topologia os algoritmos que utilizam *p-cycle* conseguem prover um maior compartilhamento de caminhos de *backup* utilizando

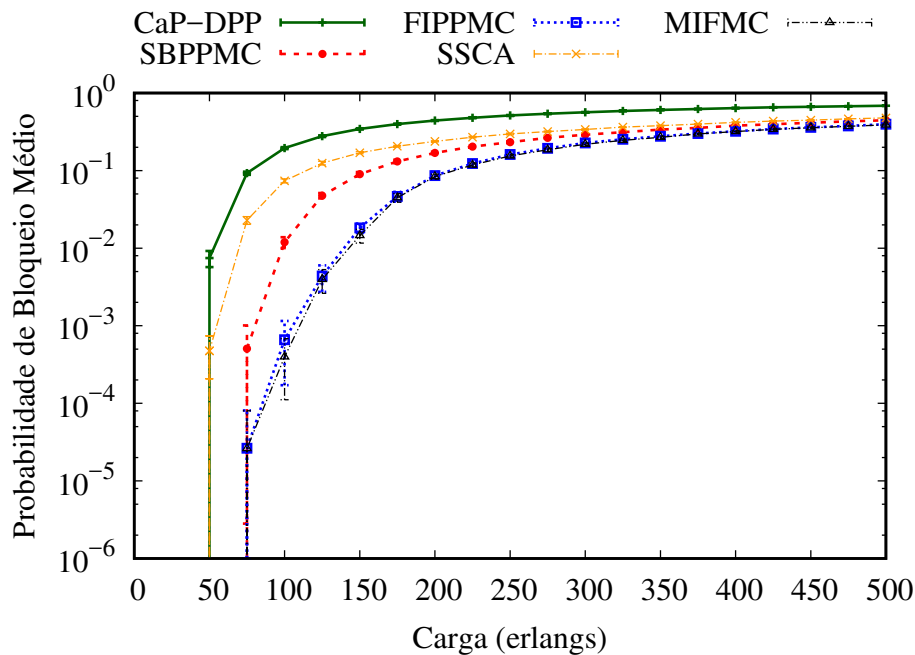


Figura 3.3: Bloqueio de banda dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.

menos recurso da rede. O algoritmo Cap-DPP produziu os valores de BBR mais alto, pois ele não usa caminhos compartilhados para prover proteção. O algoritmo SBPPMC produz valores de BBR inferiores ao do SSCA para todas as cargas simuladas, já que o algoritmo SBPPMC usa a técnica de multigrafo apresentada para geração de caminhos, gerando uma melhor distribuição de caminhos na rede. A diferença entre os valores de BBR produzido pelo FIPPMC e o MIFMC é quase uma ordem de grandeza quando comparada com o SBPPMC e quase duas ordens de magnitude quando comparada com o algoritmo SSCA, e pode chegar a quatro ordens de grandeza quando comparada com o algoritmo de Cap-DPP. Sob cargas elevadas de 300 erlangs, os algoritmos SBPPMC, Cap-DPP, FIPPMC e MIFMC produzem bloqueios semelhante.

A Figura 3.4 mostra a relação de *crossstalk* por *slot* gerado para a topologia Pan-European. Os valores de CpS gerados pelos algoritmos FIPPMC e MIFMC iniciam com um valor de 0,02 e aumentam até 0,40 e 0,42, respectivamente. Os valores de CpS gerados para os algoritmos SSCA e SBPPMC iniciam com um valor de 0,02 e aumentam até 0,17. Os valores de CpS gerado pelo algoritmo CaP-DPP iniciam em 0,02 e aumentam até 0,05. Os algoritmos FIPPMC e MIFMC produzem CpS mais altos do que os outros três algoritmos, pois esses algoritmos produzem menor bloqueio de requisições implicando na utilização de mais recursos, o que implica em uma maior probabilidade de gerar CpS. Os valores de CpS gerados pelos algoritmos FIPPMC e MIFMC começam com um valor de 0,02 e aumentam rapidamente com o aumento da carga, pois, para os algoritmos que usam *p-cycle*, uma maior conectividade de rede leva a valores de BBR mais baixo e a uma maior utilização da capacidade de rede. Embora o algoritmo SSCA produza um bloqueio maior que o algoritmo SBPPMC, ambos produzem CpS semelhante, isso acontece porque o uso do multigrafo gera caminhos menores usando menos recursos e gerando menos CpS.

A Figura 3.5 mostra a razão de fragmentação gerada para a topologia Pan-European. O

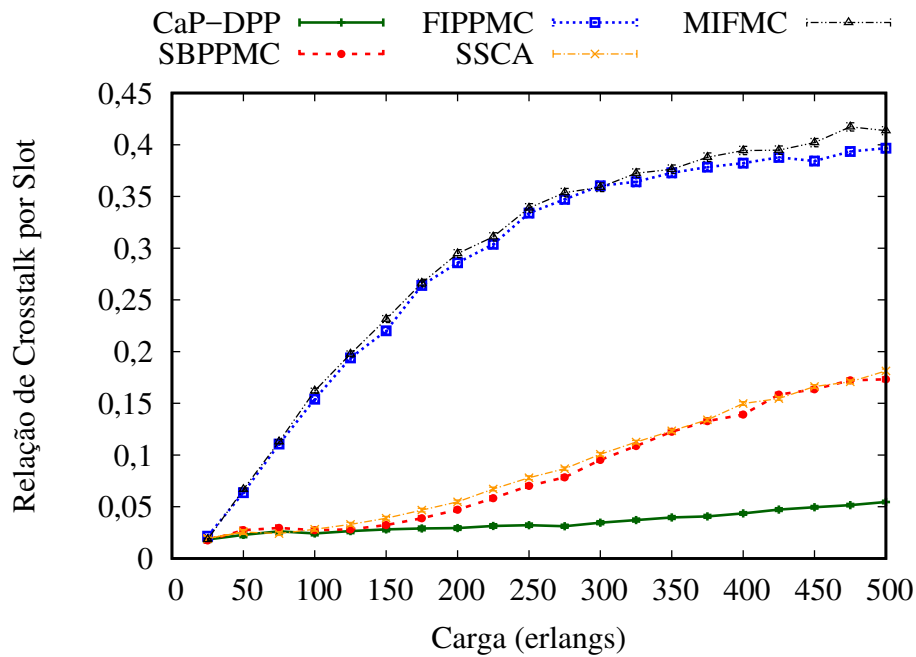


Figura 3.4: *Crosstalk* por *slot* dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.

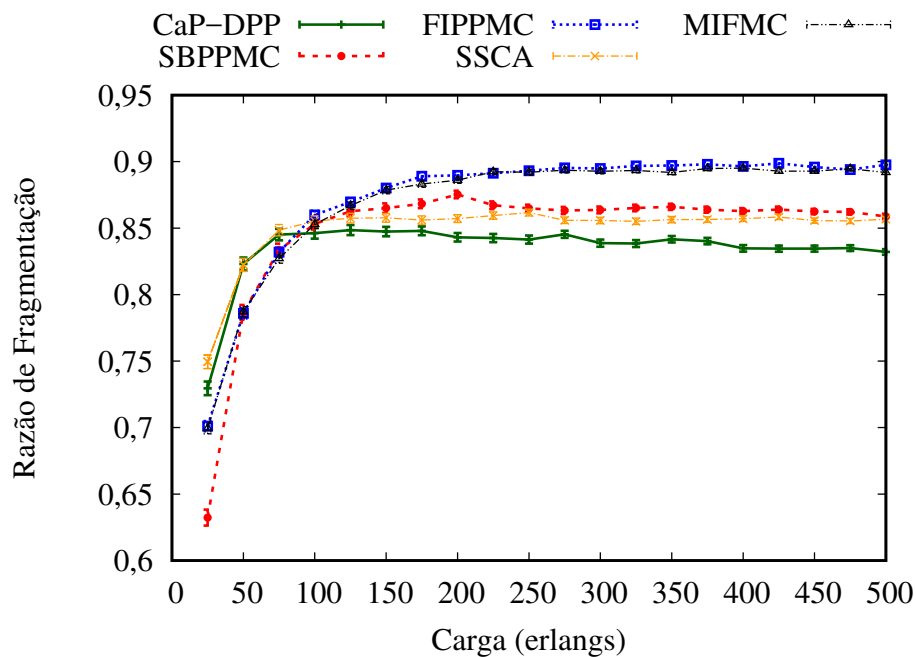


Figura 3.5: Razão de fragmentação dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.

algoritmo Cap-DPP produz uma razão de fragmentação menor que o algoritmo SSCA, isto ocorre como consequência do maior bloqueio gerado pelo algoritmo Cap-DPP fazendo com que um número menor de caminhos seja desestabelecidos e consequentemente gerando menor razão de fragmentação. O algoritmo MIFMC produz uma FR 2% menor que a dada pelo algoritmo FIPPMC apesar do MIFMC produzir menor bloqueio de requisições, isso ocorre pois maiores

caminhos tendem a ser mais compartilhados gerando um menor número de desconexões. Os algoritmos FIPPMC e MIFMC produzem uma razão de fragmentação quase 10% maior que a dada pelo SBPPMC, uma vez que mais requisições são aceitas pelos algoritmos FIPPMC e MIFMC gerando estabelecimento de conexões e conseqüentemente mais fragmentação. O algoritmo Cap-DPP produz a menor razão de fragmentação, uma vez que aceita o menor número de requisições, gerando menor número de saltos e menor razão de fragmentação.

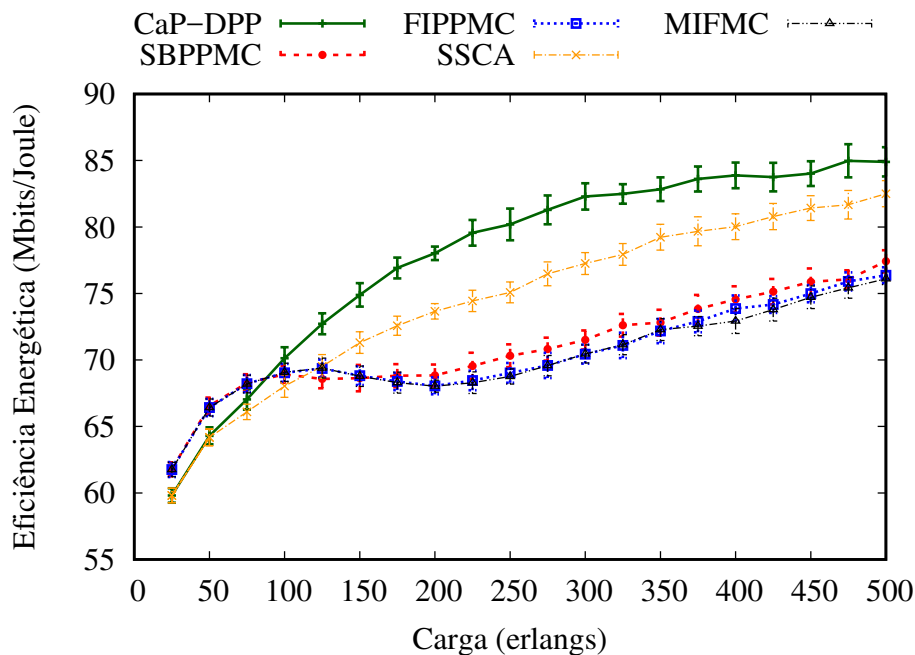


Figura 3.6: Eficiência Energética dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.

A Figura 3.6 mostra a eficiência energética gerada para a topologia Pan-European. Sob cargas mais altas, o algoritmo Cap-DPP produz uma maior eficiência energética, como consequência da alocação de caminhos com menor número de saltos, ou seja, o bloqueio produzido nesta topologia afeta a capacidade de estabelecer caminhos com comprimento arbitrário. O FIPPMC e o MIFMC produzem a menor eficiência energética, como consequência da menor BBR gerada devido a utilização de *p-cycle* FIPP e do maior número médio de saltos dos caminhos primários. Não há muita diferença entre a eficiência energética do SBPPMC e os algoritmos que utilizam o *p-cycle* FIPP. A diferença ocorre apenas sob cargas altas, uma vez que FIPPMC e MIFMC produzem um menor bloqueio sob essas cargas. O algoritmo SSCA sob cargas menores que 100 erlangs tem pouca eficiência energética, apesar de produzir maior bloqueio de requisições, como consequência do número médio de saltos dos caminhos primários.

A Figura 3.7 mostra o número médio de saltos do caminho primário gerado para a topologia Pan-European. Essa métrica expressa o impacto dos valores de BBR produzido em relação a forma de criação dos caminhos primários. A utilização do multigrafo pelos algoritmos SBPPMC, FIPPMC e MIFMC para encontrar o menor caminho primário mantém um menor número de saltos quando comparado aos algoritmos Cap-DPP e SSCA, que encontram a rota para posteriormente encontrar *slots* disponíveis obedecendo contiguidade e continuidade do espectro. A utilização do multigrafo possibilita identificar o caminho disponível considerando as restrições

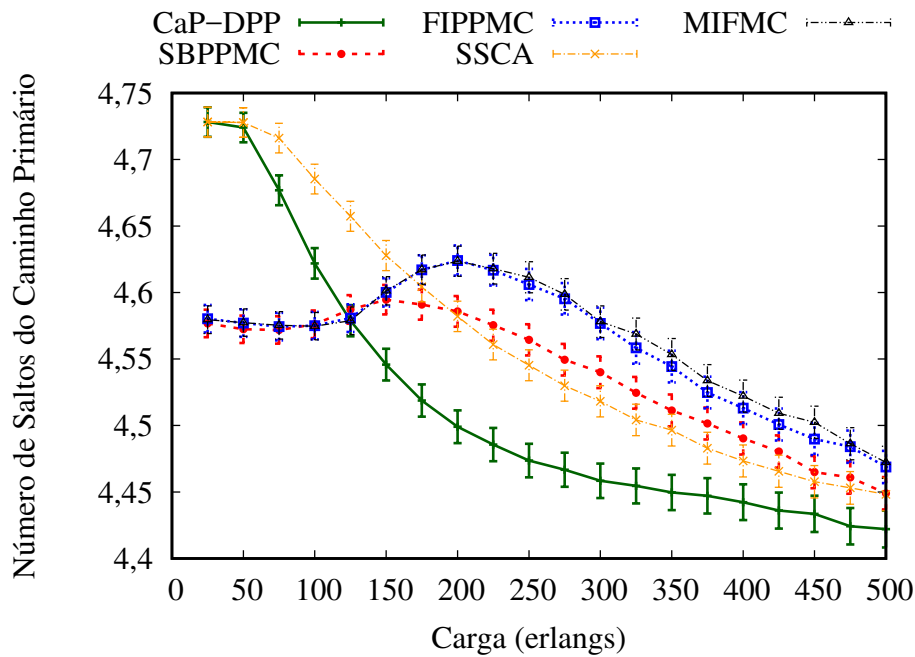


Figura 3.7: Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.

de contiguidade e continuidade do espectro. Por outro lado, os algoritmos Cap-DPP e SSCA que utilizam grafo, encontram a rota para posteriormente analisar a disponibilidade de núcleo e espectro, o que pode ocasionar na escolha de rotas indisponíveis.

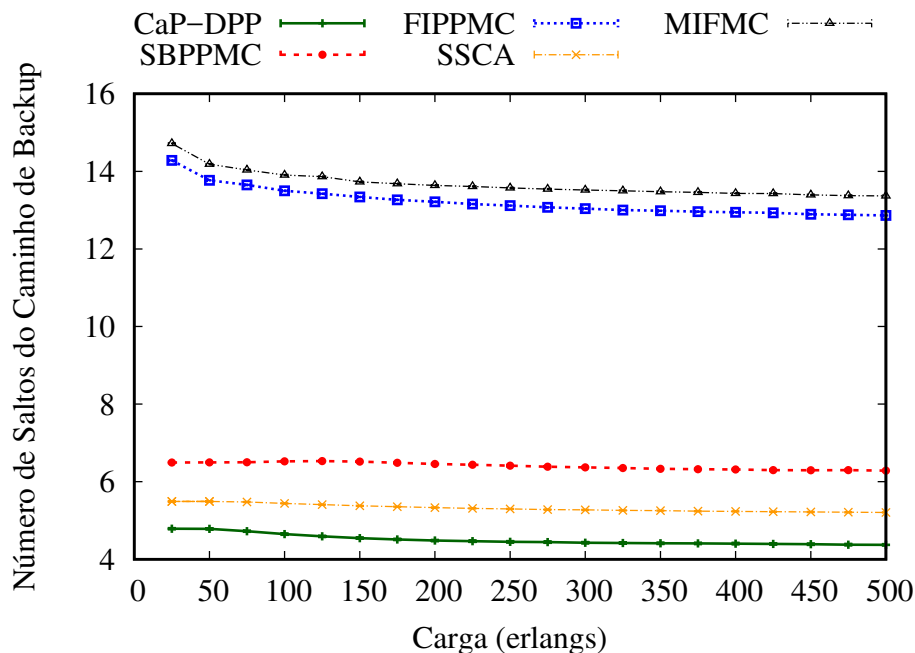


Figura 3.8: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia Pan-European.

A Figura 3.8 mostra o número médio de saltos caminho de *backup* gerado para a topologia Pan-European. Através desta métrica é possível avaliar a quantidade de recursos reservados

pelos p -cycles alocados com relação aos caminhos de *backup* compartilhados. Os algoritmos que utilizam p -cycle produzem caminhos de *backup* com três vezes mais número médio de saltos do que os caminhos que não utilizam. O algoritmo MIFMC produz caminhos de *backup* com número médio de saltos sempre maior que o algoritmo FIPPMC, isto ocorre pois o algoritmo MIFMC prioriza p -cycles transzonais ao caminho primário, e esses p -cycles tendem a ser maiores que os caminhos que usam os mesmos enlaces do p -cycle. Os algoritmos SBPPMC produz caminhos de *backup* com número médio de saltos maiores que o algoritmo SSCA, isto ocorre porque o algoritmo SSCA produz um maior bloqueio de requisições o que implica no estabelecimento de apenas caminhos menores.

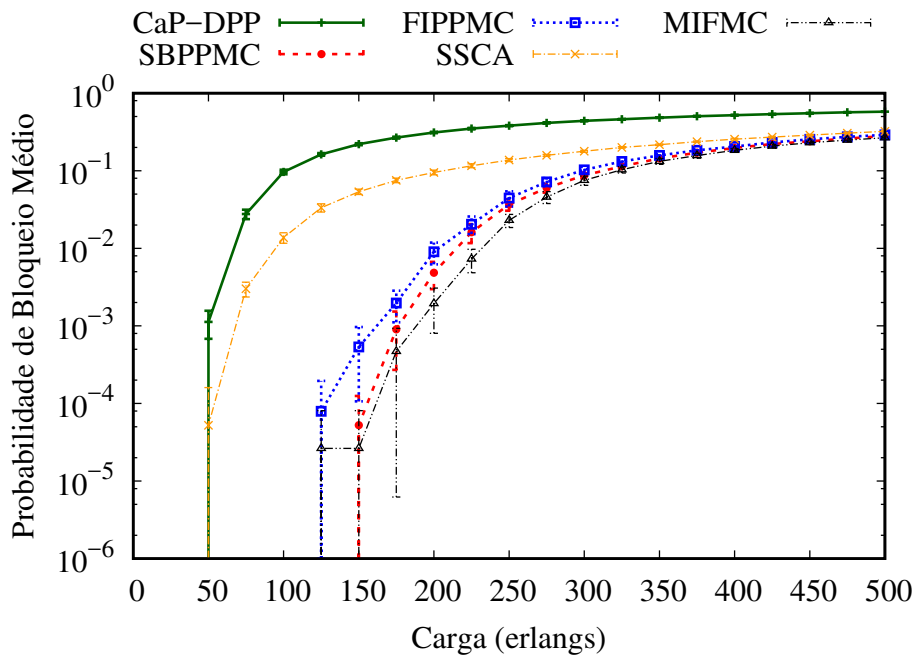


Figura 3.9: Bloqueio de banda dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.

A Figura 3.9 mostra a probabilidade de bloqueio para a topologia USA. Enquanto o Cap-DPP e o SSCA iniciam o bloqueio de requisições sob cargas de 50 erlangs, o FIPPMC e o MIFMC iniciam o bloqueio de requisições somente sob cargas de 125 erlangs e o SBPPMC, por sua vez, inicia o bloqueio de requisições apenas sob cargas de 150 erlangs. O algoritmo SBPPMC produziu a segunda menor BBR, sendo os valores de BBR produzido maiores apenas que os valores de BBR produzido pelo algoritmo MIFMC, isso acontece porque os nós da topologia USA são mais conectados que os da topologia Pan-European. O algoritmo FIPPMC produziu uma BBR maior que o MIFMC como consequência da maior conectividade dos nós nessa topologia, permitindo a alocação de p -cycles com interferência mínima pelo algoritmo MIFMC. Sob cargas de 150 erlangs, os algoritmos FIPPMC e SBPPMC produzem padrões de BBR semelhante, já que, enquanto o algoritmo SBPPMC usa menos recursos de rede, o MIFMC usa o compartilhamento dos p -cycles e a interferência mínima para evitar sobrecarga de enlaces. Sob carga de 150 erlangs, os valores de BBR produzidos pelos algoritmos MIFMC e SBPPMC são mais que três ordem de grandeza menores quando comparados com os valores de BBR produzidos pelo algoritmo CaP-DPP e mais do que duas ordem de grandeza menor quando

comparado com os valores de BBR produzidos pelo algoritmo SSCA e mais que uma ordem de magnitude menor quando comparado com os valores de BBR produzidos pelo algoritmo FIPPMC. Sob carga de 150 erlangs, o algoritmo FIPPMC produz valores de BBR duas ordens de grandeza menor que os valores produzidos pelo algoritmo CaP-DPP e quase duas ordens de grandeza menor quando comparado aos valores de BBR produzidos pelo algoritmo SSCA. Isso ocorre porque o *p-cycle* promove maior compartilhamento, gerando uma menor quantidade de caminhos de *backup* desconectados e reduzindo o número de *slots* disponível que não podem ser usados. Até 200 erlangs, o algoritmo MIFMC combina as vantagens de compartilhar *p-cycle* e interferência mínima, produzindo menos bloqueio do que os outros algoritmos avaliados.

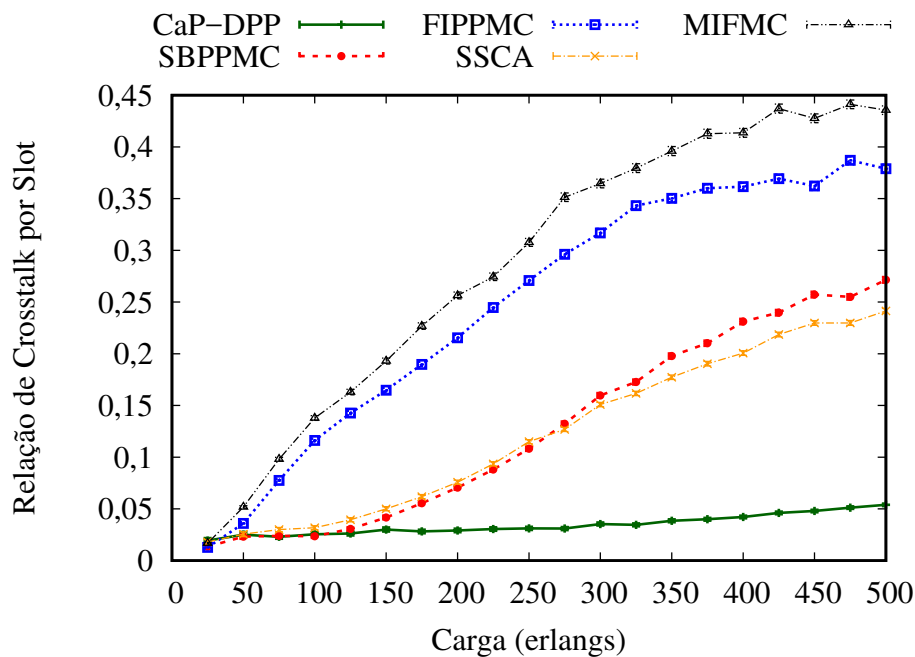


Figura 3.10: *Crosstalk* por *slot* dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.

A Figura 3.10 mostra a relação de *crosstalk* por *slot* gerado para a topologia USA. Os valores de CpS gerados para os algoritmos FIPPMC e MIFMC iniciam em um valor de 0,02 e aumentam até 0,38 e 0,44, respectivamente. Os valores de CpS gerados para os algoritmos SSCA e SBPPMC iniciam em um valor de 0,02 e aumentam até 0,24 e 0,26, respectivamente. Tal como na topologia Pan-European, os algoritmos FIPPMC e MIFMC produzem CpS mais elevados que os outros três algoritmos. O algoritmo MIFMC produz CpS maior que o algoritmo FIPPMC, como consequência da alta utilização da capacidade da rede, gerada pelo maior número de requisições aceitas. Embora o algoritmo SBPPMC produza valores de BBR próximos aos produzidos pelos algoritmos FIPPMC e MIFMC, o CpS do algoritmo SBPPMC é menor, pois os algoritmos com *p-cycle* utilizam mais recursos de rede. Os algoritmos SBPPMC e SSCA produzem CpS próximos, porém, o SBPPMC produz valores de BBR menores, devido a melhor distribuição na alocação feita pelo uso do multigrafo.

A Figura 3.11 mostra a razão de fragmentação gerada para a topologia USA. O algoritmo Cap-DPP produz uma razão de fragmentação 2% menor que a dada pelo algoritmo SSCA, principalmente para cargas mais altas, como consequência da média de número de saltos gerada

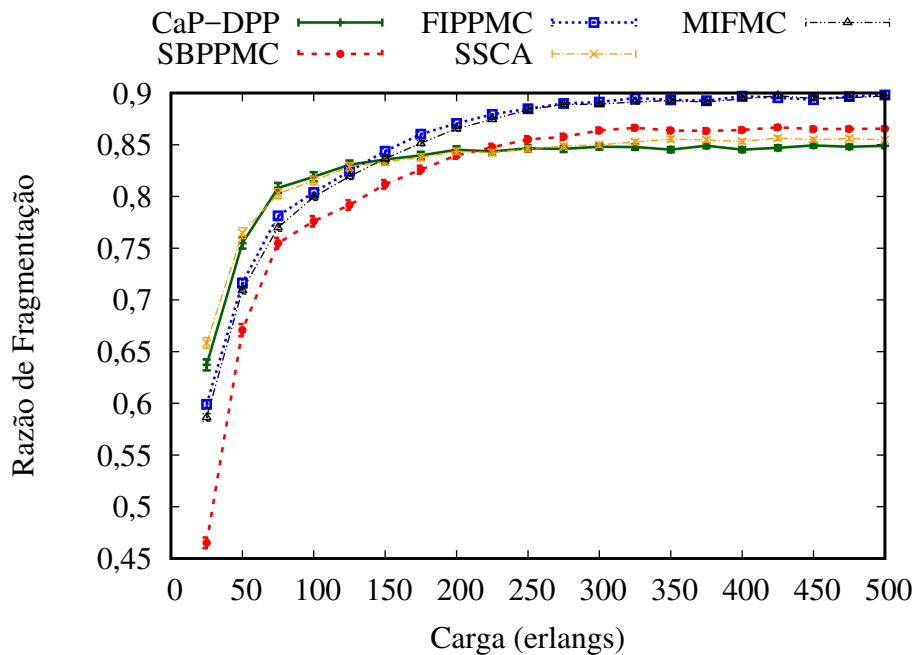


Figura 3.11: Razão de fragmentação dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.

pele maior bloqueio produzido pelo Cap-DPP. O maior bloqueio de requisições do Cap-DPP aceita apenas caminhos menores, e conseqüentemente utiliza menos *slots* e gera uma menor fragmentação. O algoritmo FIPPMC produz uma proporção de fragmentação quase 1% superior à dada pelo algoritmo MIFMC, como consequência do uso de *p-cycle* transzonais ao caminho primário, o que permite maior compartilhamento, e conseqüentemente, conseqüentemente menos desconexão e menos fragmentação. Os algoritmos FIPPMC e MIFMC produzem uma taxa de fragmentação 8% maior que a dada pelo SBPPMC, uma vez que o algoritmo SBPPMC gera menos fragmentação no caminho de *backup* devido ao seu menor número de saltos.

A Figura 3.12 mostra a eficiência energética gerada para a topologia USA. Até cargas de 100 erlangs, não há muita diferença entre a eficiência energética do Cap-DPP e a dos outros algoritmos. A diferença ocorre apenas sob cargas altas, como consequência dos algoritmos FIPPMC, MIFMC e SBPPMC produzirem valores de BBR significativamente menores sob essas cargas. Sob cargas elevadas, o Cap-DPP produz maior eficiência energética do que o algoritmo SSCA, uma vez que o Cap-DPP produz maior bloqueio. Sob cargas superiores a 175 erlangs, a eficiência energética para o algoritmo SSCA é maior que os algoritmos FIPPMC, SBPPMC e MIFMC, pois, sob essas cargas, o algoritmo SSCA produz maiores valores de BBR. Não há muita diferença entre a eficiência energética do SBPPMC e os algoritmos que utilizam *p-cycle*, isso ocorre devido ao uso do multigrafo para encontrar o caminho primário, gerando caminhos de forma similar.

A Figura 3.13 mostra o número médio de saltos do caminho primário gerado para a topologia USA. Os algoritmos SBPPMC, FIPPMC e MIFMC, que utilizam multigrafo, produzem resultados semelhantes para qualquer carga simulada. Sob cargas menores, os algoritmos que utilizam multigrafo produzem um menor número de saltos que os algoritmos que não utilizam. Isto ocorre pois sob cargas menores todos os algoritmos produzem bloqueio de requisições si-

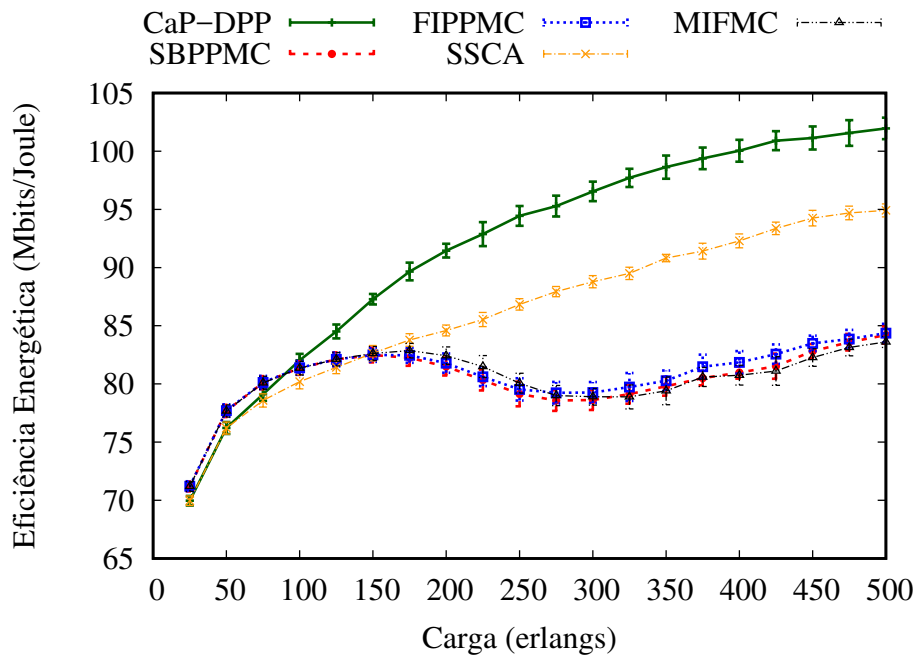


Figura 3.12: Eficiência Energética dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.

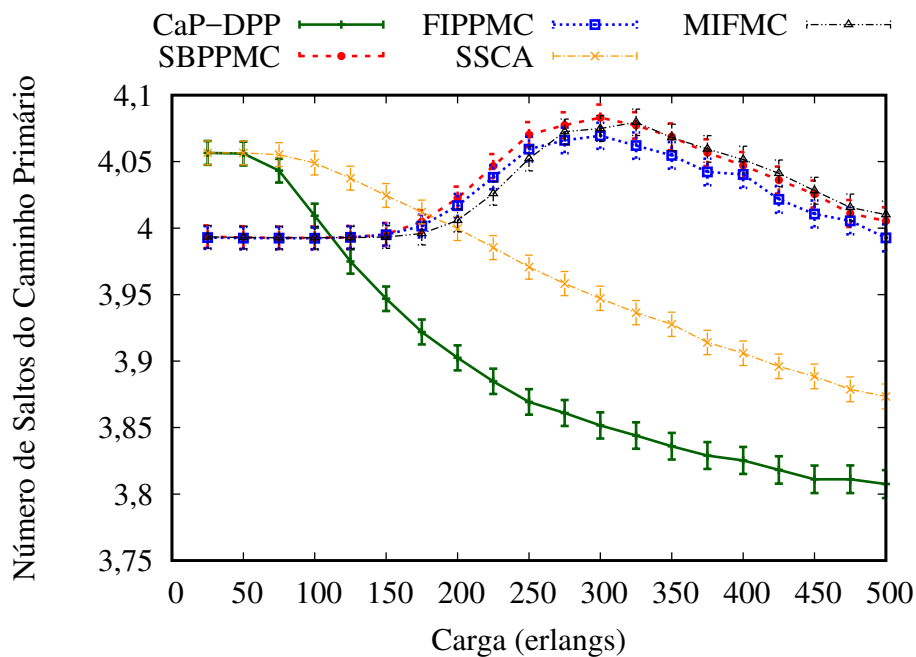


Figura 3.13: Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.

milar. No entanto, conforme a carga da rede aumenta, aumenta também os valores de BBR dos algoritmos da rede que não utilizam multigrafo. O maiores valores de BBR dos algoritmos Cap-DPP e SSCA fazem com que esses algoritmos gerem caminhos mais curtos, visto que, com a maior BBR, alguns enlaces da rede são sobrecarregados, impossibilitando a alocação de caminhos maiores.

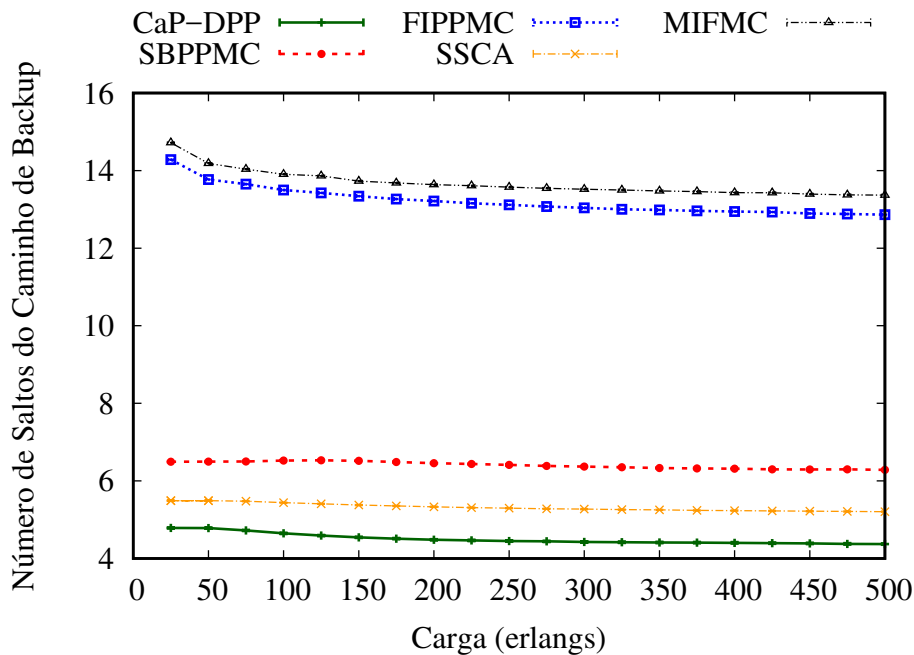


Figura 3.14: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia USA.

A Figura 3.14 mostra o número médio de saltos do caminho de *backup* gerado para a topologia USA. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com quase três vezes mais número médio de saltos que os caminhos que não utilizam. Para topologia Pan-European, o algoritmo MIFMC produz caminhos de *backup* com número médio de saltos maior do que os produzidos pelo algoritmo FIPPMC, isto ocorre pois o algoritmo MIFMC prioriza *p-cycles* transzonais, e esses *p-cycles* tendem a ser maiores do que os caminhos que usam os mesmos enlaces do *p-cycle*. O algoritmo Cap-DPP, por sua vez, produz o menor número de saltos para os caminhos de *backup*, como consequência dos maiores valores de BBR produzidos. Por fim, para topologia Pan-European, o algoritmo SBPPMC aloca caminhos de *backup* com uma salto maior que o algoritmo SSCA isto ocorre devido ao menor bloqueio gerado pelo algoritmo SBPPMC, o que ocasiona em menos recursos disponíveis.

A Figura 3.15 mostra a probabilidade de bloqueio para a topologia NSF. Enquanto os algoritmos Cap-DPP e SSCA iniciam o bloqueio de requisições sob cargas de 25 e 50 erlangs, respectivamente, enquanto os algoritmos FIPPMC e MIFMC iniciam o bloqueio de requisições sob cargas de 125 erlangs e o SBPPMC inicia o bloqueio de requisições apenas sob cargas de 150 erlangs. Os valores de BBR produzidos pelo algoritmo SBPPMC são sempre inferiores aos dos outros algoritmos simulados, evidenciando o benefício de se criar caminhos de *backup* compartilhados usando-se multigrafos para topologias com menos conectividade. Sob carga de 150 erlangs, a BBR produzida pelo algoritmo SBPPMC é uma ordem de magnitude menor quando comparado com as BBRs produzidas pelos algoritmos FIPPMC e MIFMC, desde que a baixa conectividade da topologia NSF torna difícil criar *p-cycles*. Os algoritmos FIPPMC e MIFMC produzem resultados mais semelhantes na topologia NSF que na topologia USA, uma vez que a baixa conectividade da topologia NSF dificulta a criação de *p-cycles* usando interferência mínima, de modo que os algoritmos FIPPMC e MIFMC criam caminhos de *backup*

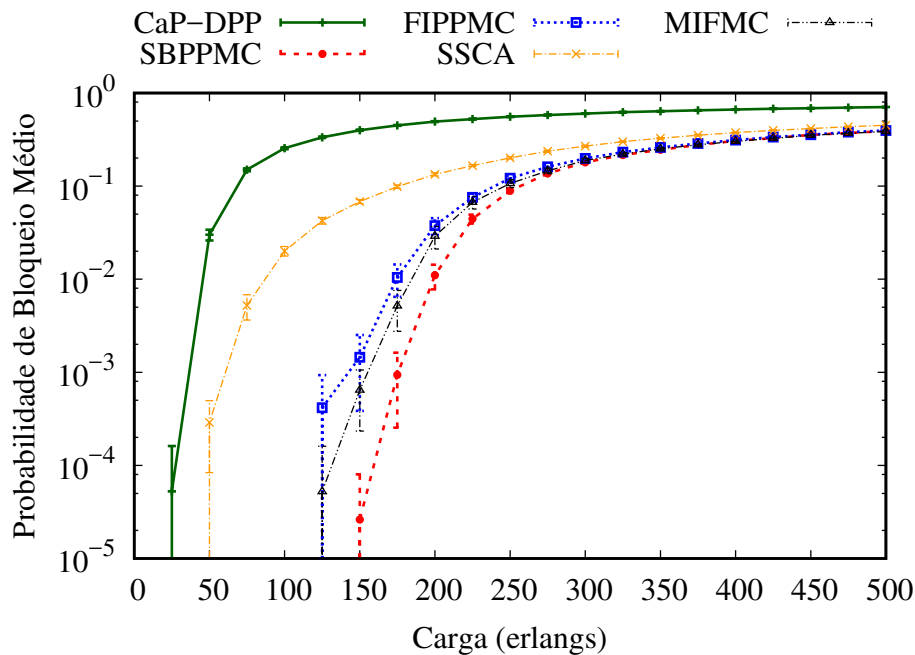


Figura 3.15: Bloqueio de banda dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.

semelhantes. Sob carga de 125 erlangs, o algoritmo MIFMC produz valores de BBR quase uma ordem de magnitude menor quando comparado com os valores de BBR produzidos pelo algoritmo FIPPMC, evidenciando os benefícios de usar interferência mínima. O algoritmo CaP-DPP mantém os valores de BBR mais altos para todas as topologias avaliadas porque não usa o compartilhamento no caminho de *backup* e também devido à forma de escolha dos caminhos utilizados.

A Figura 3.16 mostra a razão de *crossstalk* por *slot* gerado para a topologia NSF. Os valores de CpS gerados para os algoritmos FIPPMC e MIFMC iniciam sob cargas baixas com um valor de 0,02 e aumentam até 0,56 e 0,57, respectivamente, conforme o acréscimo da carga. Os valores de CpS gerados para os algoritmos SSCA e SBPPMC iniciam com um valor de 0,02 e aumentam até 0,43 e 0,51, respectivamente. Apesar do algoritmo SBPPMC produzir baixo bloqueio e alta utilização, ele também produz baixo CpS. A utilização do algoritmo SBPPMC para topologias com menos conectividade produz menos *crossstalk* do que os algoritmos FIPPMC e MIFMC, uma vez que os algoritmos que usam o *p-cycle* utilizam mais recurso da rede. O algoritmo SSCA produz valores de CpS mais próximos àqueles gerados pelos algoritmos FIPPMC, MIFMC e SBPPMC que o Cap-DPP, uma vez que ele bloqueia menos requisições, consequentemente usa mais recursos da rede. O algoritmo CaP-DPP produz o menor CpS devido ao alto bloqueio produzido.

A Figura 3.17 mostra a razão de fragmentação gerada para a topologia NSF. Até cargas de 200 erlangs, o algoritmo SSCA produz menor taxa de fragmentação do que o algoritmo Cap-DPP, devido ao compartilhamento de caminho de backup do algoritmo SSCA, que gera menos desconexão e, consequentemente, menor fragmentação. Sob cargas elevadas, o algoritmo Cap-DPP produz menor razão de fragmentação do que o SSCA, devido à maior BBR gerado pelo Cap-DPP. O algoritmo FIPPMC produz uma taxa de fragmentação quase 1% maior que a dada

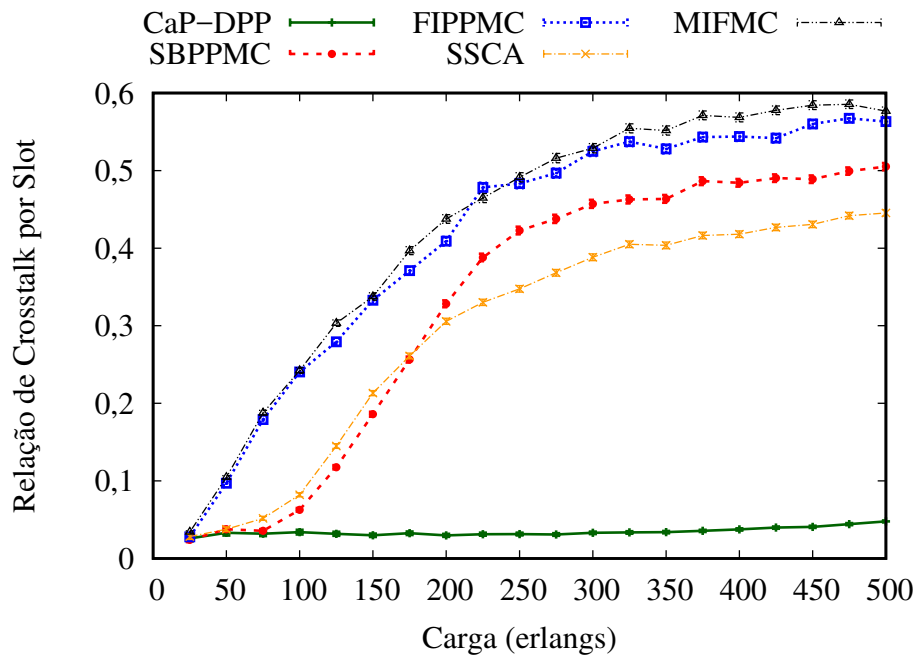


Figura 3.16: *Crosstalk* por *slot* dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.

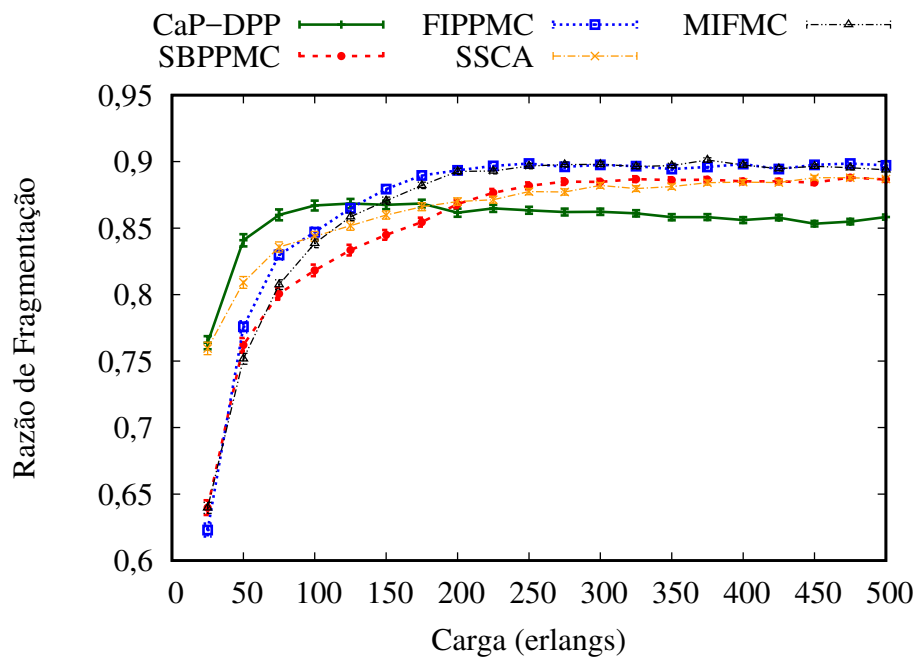


Figura 3.17: Razão de fragmentação dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.

pelo MIFMC, apesar de que o algoritmo FIPPMC produz maior BBR. Isso acontece devido a maior utilização de enlaces pelos *p-cycles* gerados pelo MIFMC, produzindo maior compartilhamento e sendo estabelecido um número menor de vezes. O algoritmo SBPPMC produz uma taxa de fragmentação 8% inferior à dada pelo FIPPMC apesar do algoritmo SBPPMC produzir uma menor BBR, isto ocorre pois algoritmos que utilizam *p-cycle* além de gerar fragmentação

para o caminho primário geram uma maior fragmentação para os caminhos de *backup*, visto que o *p-cycle* gerado pelo algoritmo FIPPMC utiliza duas vezes mais enlace do que os enlaces utilizados pelo caminho de *backup* gerado pelo algoritmo SBPPMC.

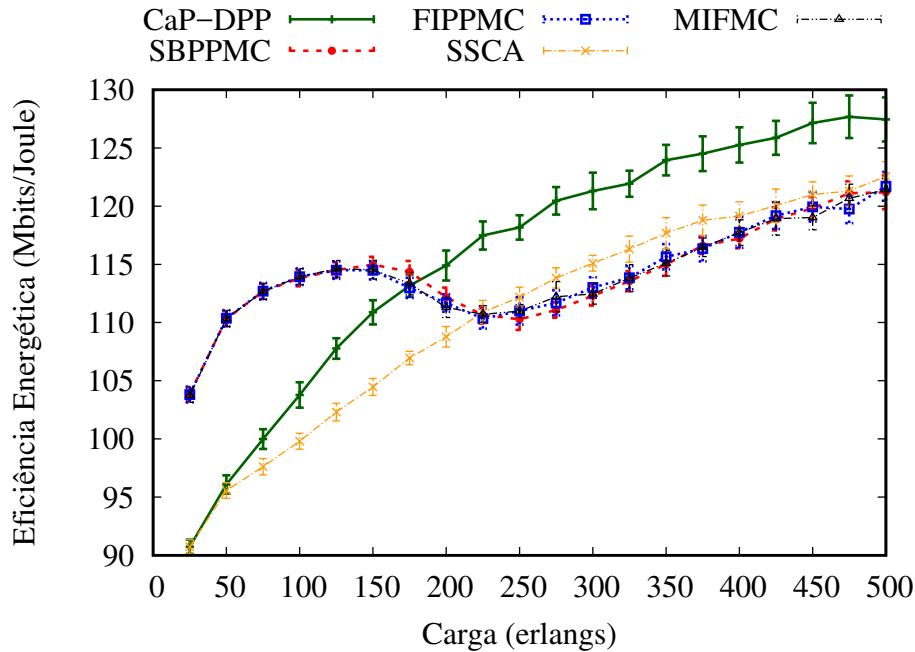


Figura 3.18: Eficiência Energética dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.

A Figura 3.18 mostra a eficiência energética gerada para a topologia NSF. Até cargas de 175 erlangs, os algoritmos SBPPMC, FIPPMC e MIFMC produzem maior eficiência energética do que os algoritmos SSCA e Cap-DPP, apesar de SBPPMC, FIPPMC e MIFMC produzirem taxas de bloqueio significativamente menores sob essas cargas. Sob cargas mais elevadas, o algoritmo Cap-DPP produz maior eficiência energética do que o algoritmo SSCA, uma vez que o Cap-DPP produz um maior bloqueio de requisições. Ainda sob carga altas, o algoritmo SSCA produz eficiência de energia próximas aos algoritmos SBPPMC, FIPPMC e MIFMC, apesar de SBPPMC, FIPPMC e MIFMC produzirem taxas de bloqueio significativamente menores sob essas cargas. Sob cargas inferiores a 150 erlangs, o SBPPMC, FIPPMC e MIFMC têm um consumo de energia semelhante. Sob cargas maiores que 225 erlangs, embora o algoritmo SSCA produza um BBR maior que os algoritmos FIPPMC, SBPPMC e MIFMC, estes algoritmos produzem eficiência energética com valores muito próximos, evidenciando o benefício do uso do multigrafo.

A Figura 3.19 mostra o número médio de saltos do caminho primário gerado para a topologia USA. Como nas topologias anteriores, os algoritmos SBPPMC, FIPPMC e MIFMC, que utilizam o multigrafo, alocam caminhos primários com número médio de saltos semelhantes para qualquer carga simulada. Além disso, os algoritmos que utilizam o multigrafo alocam caminhos sempre com menor número médio de saltos do que os caminhos que não utilizam o multigrafo. Os algoritmos Cap-DPP e SSCA alocam caminhos com número médio de saltos semelhantes em comparação aos outros algoritmos. No entanto, o número médio de saltos do caminho primário gerado pelo Cap-DPP é sempre menor que o gerado pelo SSCA.

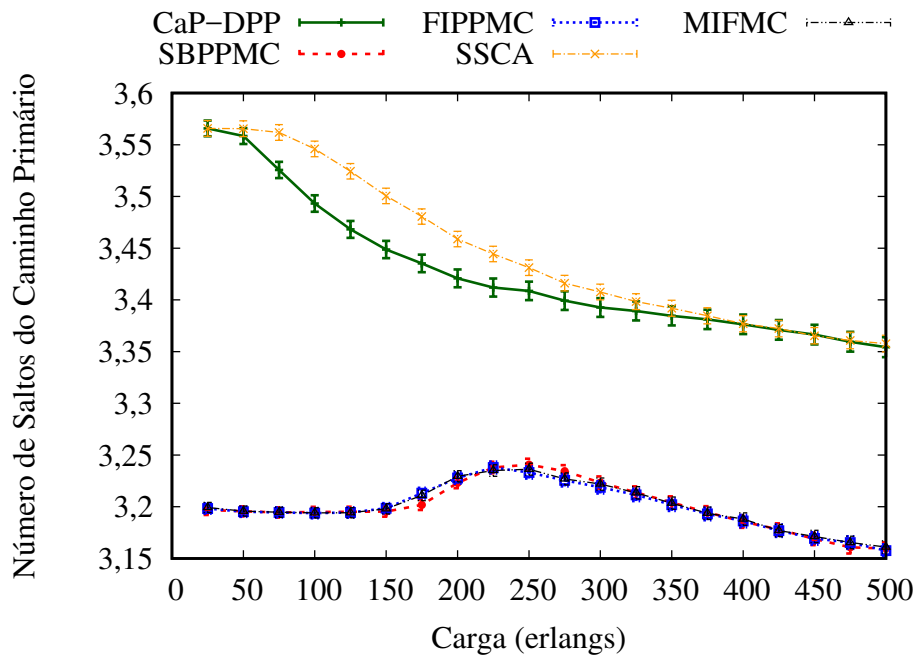


Figura 3.19: Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.

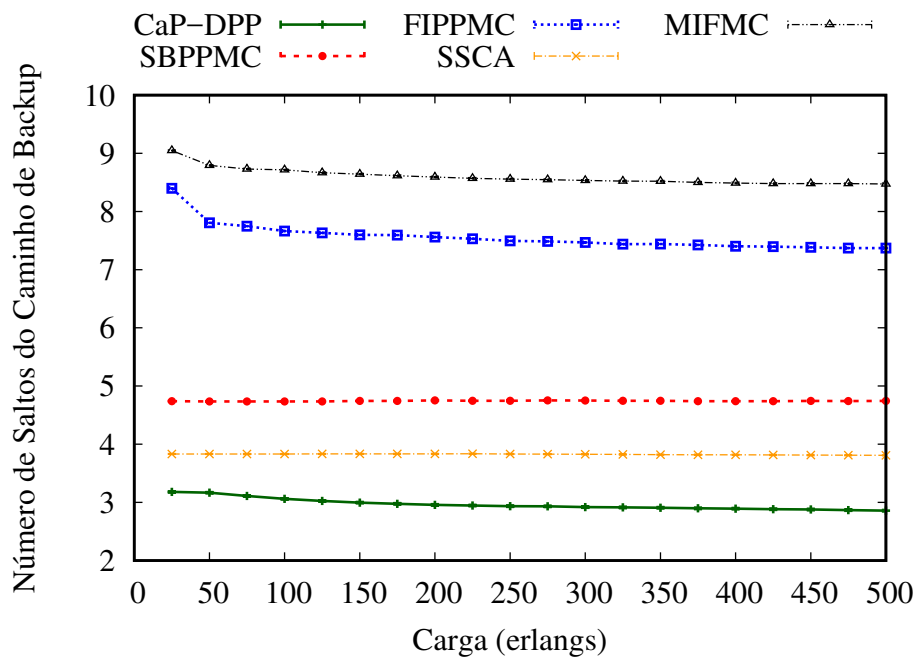


Figura 3.20: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPP, SSCA, SBPPMC, FIPPMC e MIFMC em função da carga da rede para topologia NSF.

A Figura 3.20 mostra o número médio de saltos do caminho de *backup* gerado para a topologia NSF. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com quase duas vezes mais número médio de saltos que os caminhos que não o utilizam. Os algoritmos FIPPMC e MIFMC produziram número de saltos no caminho de *backup* com resultados significativos em relação às outras topologias. A menor conectividade da topologia NSF influenciou

os tamanhos dos *p-cycles* criados, de forma que o algoritmo MIFMC alocou em média uma unidade a mais em número de saltos do que o alocado pelo algoritmo FIPPMC. O número de saltos do caminho de *backup* para os algoritmo SBPPMC é uma unidade maior que o número de saltos do caminho de *backup* do algoritmo SSCA, pois com o maior número de requisições aceitas pelo algoritmo SBPPMC, aumenta-se a dificuldade de criação de caminhos de *backup*, sendo necessário um maior número de saltos para prover proteção.

3.8 Síntese do capítulo

Neste capítulo, apresentaram-se os algoritmos SBPPMC, FIPPMC e MIFMC para proteção de redes ópticas elásticas com multiplexação espacial. Os três algoritmos utilizam multigrafo para encontrar caminhos através de topologias com múltiplos núcleos. A utilização do multigrafo proporcionou um menor número médio de saltos aos algoritmos que o utilizam, para todas as topologias. Ao encontrar caminhos utilizando-se o multigrafo, leva-se em consideração não somente os enlaces da rede, como também, os slots disponíveis, a contiguidade e continuidade de núcleo e espectro e o *crosstalk* atual do enlace a ser utilizado.

Além de introduzir os três algoritmos, este capítulo comparou ainda técnicas de proteção, tais como: caminhos compartilhados, *p-cycle* FIPP e *p-cycle* FIPP com interferência mínima. Os resultados mostraram que o uso de multigrafo é uma boa escolha de estrutura para criação de caminhos de *backup*, uma vez que os algoritmos FIPPMC, MIFMC e SBPPMC que o utilizam, produzem bloqueio aceitável para redes ópticas elástica com multiplexação espacial, apesar da reserva de largura de banda para pré-provisionamento de caminhos de *backup*. Além disso, o uso do *p-cycle* FIPP é mais indicado para redes com maior conectividade dos nós, uma vez que os algoritmos que usam o *p-cycle* FIPP produzem BBR mais baixo em redes com essas características.

Em geral, os gráficos mostraram que quanto menor for o BBR, maior será o CpS. Além disso, o BBR alto ao usar o algoritmo com proteção 1:1 (Cap-DPP), faz com que o CpS permaneça baixo para todas as topologias. O efeito da alta utilização da capacidade da rede afeta claramente o CpS. Pode-se ver que o CpS médio está correlacionado com a BBR com exceção dos algoritmos que usam *p-cycle* devido a sua alta taxa de compartilhamento, que gera uma economia nos recursos de *backup*.

Em geral, os algoritmos que produzem maior BBR tendem a produzir maior eficiência energética do que outros algoritmos avaliados, uma vez que apenas caminhos menores são estabelecidos. No entanto, quando os multigrafos são utilizados é possível aumentar a eficiência energética, devido a melhor distribuição na alocação de caminhos.

Capítulo 4

Uso de diferentes formatos de modulação em redes ópticas elásticas com multiplexação espacial e com proteção de caminho

A modulação adaptativa é um termo utilizado para soluções que definem a modulação baseada em alguma métrica de QoT. Nesta Tese, o formato de modulação a ser utilizado nos caminhos levará em consideração a distância entre os nós de origem e destino (alcance óptico). A transmissão adaptativa à distância (DAT) escolhe o formato de modulação considerando apenas a distância de transmissão [18, 20]. Neste método de modulação adaptativa, o formato de modulação mais eficiente, do ponto de vista espectral, é selecionado de forma que o comprimento do caminho não exceda o alcance da transmissão.

O comportamento da relação entre eficiência óptica e alcance de transmissão é exemplificado na Figura 4.1. Na figura, a eficiência óptica refere-se a quantidade de bits/símbolo transmitida ao se utilizar um determinado formato de modulação. Por outro lado, quanto maior a quantidade de bits/símbolo menor deve ser a distância entre os nós de origem e destino (alcance óptico em quilômetros).

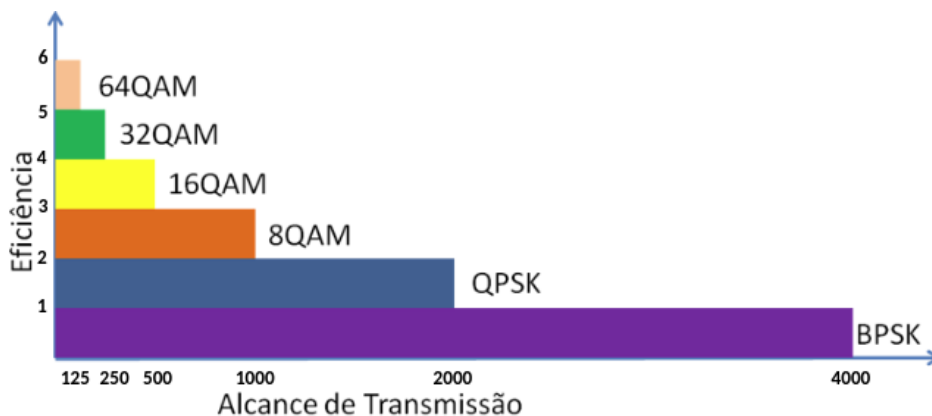


Figura 4.1: Eficiência x Alcance de transmissão.

Neste presente trabalho, utilizaram-se formatos de modulação conforme a Tabela 4.1, que mostra a capacidade dos *slots* de acordo com o formato de modulação [104]. E este depende da distância entre os nós de origem e destino. Os valores de XT aceitável para cada formato de modulação foram retirados de [91]. Assim como no Capítulo 3, os algoritmos propostos utilizam a técnica do multigrafo. No entanto, o valor de XT aceitável é alterado conforme a modulação utilizada pelo slot e seus slots adjacentes, já que um slot impacta no XT dos slots adjacentes.

Tabela 4.1: Características de Modulação

Formato de Modulação	Bits por Símbolo	Capacidade do Slot (Gb/s)	Distancia Máxima (km)	XT Aceitável
64QAM	6	75	125	-32 dB
32QAM	5	62.5	250	-28 dB
16QAM	4	50	500	-24 dB
8QAM	3	37.5	1000	-21 dB
QPSK	2	25	2000	-18 dB
BPSK	1	12.5	4000	-16 dB

Neste capítulo, avalia-se o impacto de três novos algoritmos para proteção em redes ópticas elásticas com multiplexação espacial utilizando modulação adaptativa. Os algoritmos apresentados estendem os algoritmos apresentados no capítulo 3, adicionando modulação adaptativa. Os três algoritmos apresentados neste capítulo diferem entre si apenas na forma de proteção realizada. O objetivo dos algoritmos propostos é reduzir a probabilidade de bloqueio e garantir proteção das requisições, evitando a degradação da qualidade da transmissão no estabelecimento de novas conexões. Os algoritmos propostos definem duas rotas do nó de origem para o nó de destino, selecionando os respectivos formatos de modulação para alcançar a distância desejada. Além disso, escolhem uma banda de espectro livre (isto é, um conjunto de *slots* de frequências contíguas) na rota definida para estabelecer o circuito óptico. A qualidade dos algoritmos propostos determina o número de circuitos que podem ser estabelecidos em uma determinada topologia de rede protegida.

Para avaliar a efetividade dos algoritmos propostos, foram realizadas simulações utilizando as três topologias apresentadas no capítulo 3. Os resultados obtidos mostram-se satisfatórios, gerando uma maior aceitação de requisições que outros algoritmos com os quais foram comparados. Além disso, os algoritmos propostos possuem diferentes comportamentos, associados à topologia utilizada.

Este capítulo está organizado conforme segue. Na Seção 4.1, apresenta-se uma revisão bibliográfica. Na seção 4.2, expõe-se a notação utilizada nos algoritmos introduzidos nesse capítulo. Na Seção 4.3, introduz-se o algoritmo BARTMAN. Na Seção 6, apresenta-se o algoritmo PERFECTA. Na Seção 4.5, apresenta-se o algoritmo MMIFMC. Na Seção 4.6, compara-se o desempenho dos algoritmos propostos e outros algoritmos. Finalmente, na Seção 4.7, apresenta-se a síntese do capítulo.

4.1 Trabalhos relacionados

Nesta seção, abordam-se os principais trabalhos que utilizam modulação adaptativa para redes ópticas elásticas; no entanto, nenhum dos trabalhos apresentados trata de modulação adaptativa para EONs SDM.

Os autores em [13] avaliaram a eficiência espectral do *p-cycle* FIPP em redes ópticas elásticas convencionais juntamente com modulação adaptativa. Foram utilizados cenários estáticos, bem como, dinâmicos para avaliação do esquema.

A capacidade das tecnologias de redes ópticas e eletrônicas para melhorar a eficiência energética da Internet foi comparada em [43].

Propuseram em [47] uma solução de roteamento, nível de modulação e atribuição de espectro usando *k*-caminhos mais curtos para calcular rotas e uma política para alocar espectro, aplicando o menor *slot* inicial disponível no espectro em redes ópticas elásticas. A modulação é escolhida com base no comprimento dos caminhos, de forma que utilize uma faixa menor do espectro e que possa ser descodificada com êxito no destino.

Em [60], propõe-se um algoritmo de roteamento, nível de modulação e atribuição de espectro para redes ópticas elásticas convencionais considerando o consumo de energia do caminho primário.

Os autores em [18] fazem uma abordagem para o problema de roteamento, modulação e alocação de espectro (RMLSA) através do uso de um esquema de modulação adaptativa à distância que permite o roteamento do tráfego através de múltiplos saltos em uma topologia virtual, permitindo a continuidade do espectro e as restrições de distância de transmissão.

Em [28], propõe-se um algoritmo Routing, Modulation Level and Spectrum Assignment (RMLSA) que considera os efeitos da camada física em redes ópticas elásticas transparentes. O objetivo do algoritmo é reduzir a probabilidade de bloqueio do circuito causada pela degradação da qualidade da transmissão no estabelecimento de novos circuitos. O algoritmo proposto utiliza *k* menores caminhos para encontrar a rota.

Em [113], um esquema de roteamento dinâmico, nível de modulação e alocação de espectro (RMLSA) para as redes elásticas é proposto. O esquema considera a conversão do formato de modulação nos nós intermediários. A influência do roteamento para o surgimento da fragmentação é quantificada para decidir qual rota dos *k* menores caminhos usar. Além disto, mostra-se um cenário de recuperação de falhas para a rede simulada.

Foi abordado em [38] o problema da localização e utilização do regenerador de sinais em uma rede óptica elástica protegida. A técnica de proteção de caminho de *backup* compartilhado (SBPP) é empregada. Foi proposto também um algoritmo de regeneração de sinal.

A Tabela 4.2 apresenta uma comparação entre os trabalhos relacionados e os algoritmos propostos neste capítulo. Os trabalhos foram categorizados com relação à proteção, ao uso de multiplexação de divisão espacial, ao uso de roteamento, nível de modulação e alocação de espectro, ao uso de roteamento, nível de modulação e alocação de núcleo e espectro, e o uso do *p-cycle* para fornecer proteção e, por fim, ao uso de proteção utilizando caminhos de *backup* compartilhados.

Tabela 4.2: Comparação dos trabalhos relacionados com os algoritmos BARTMAN, PERFECTA, MMIFMC.

Abordagem	Proteção	SDM	RMLSA	RMLCSA	p -Cycle	SBPP
Chen <i>et al.</i> [13]	Sim	Não	Sim	Não	Sim	Não
Hinton <i>et al.</i> [43]	Não	Não	Não	Não	Não	Não
Jinno <i>et al.</i> [47]	Não	Não	Sim	Não	Não	Não
Moura <i>et al.</i> [60]	Não	Não	Sim	Não	Não	Não
Costa <i>et al.</i> [18]	Não	Não	Sim	Não	Não	Não
Fontinele <i>et al.</i> [28]	Não	Não	Sim	Não	Não	Não
Yin <i>et al.</i> [113]	Não	Não	Sim	Não	Não	Não
Guo <i>et al.</i> [38]	Sim	Não	Sim	Não	Não	Sim
BARTMAN	Sim	Sim	Não	Sim	Não	Sim
PERFECTA	Sim	Sim	Não	Sim	Sim	Não
MMIFMC	Sim	Sim	Não	Sim	Sim	Não

4.2 Notação

A notação matemática introduzida a seguir será usada neste capítulo. A notação adiciona a notação do capítulo anterior (Capítulo 3). As principais diferenças encontram-se nos aspectos que tratam de modulação:

$m = 1 \dots M$: formatos de modulação (Tabela 4.1);

b_m : demanda de largura de banda em *slots* levando em conta o formato de modulação a ser adotado;

$\tilde{G}_{n,b_m} = (\tilde{V}, \tilde{E}, \tilde{W})$: o n -ésimo grafo marcado tal que \tilde{E} é o conjunto de arestas conectando $\{\tilde{u}, \tilde{v}\} \in \tilde{V}$ e \tilde{W} é o conjunto de custos associados a \tilde{E} . A aresta \tilde{E} corresponde ao mapeamento de b_m aresta em G iniciando na n -ésima aresta;

$\tau(G, C, b_m) = \{\tilde{G}_{n,b_m}\}$: função que produz todos os σ grafos de G ;

$\sigma = |\{\tilde{G}_{n,b_m}\}| = C \times (N - b_m + 1)$: número de grafos extraído do multigrafo;

P_n : sequência de \tilde{G}_{n,b_m} tal que o nó fonte s é o menor nó ordenado e d é o maior nó ordenado;

$W(P_n)$: o peso do caminho P_n , a soma dos pesos de todas as arestas na sequência;

$W_{P_{s,d}}$ = peso do menor caminho entre s e d ;

Z_n : caminho de *backup* contendo os vértices u e v e arestas correspondendo ao mapeamento de b_m arestas no multigrafo G ;

Z : conjunto de todos os caminhos de *backup* ativos;

P_{Z_n} : conjunto de todos os caminhos protegidos pelo caminho de *backup* Z_n ;

$W(Z_n)$: o peso do caminho de *backup* Z_n (a soma dos pesos de todas as arestas da sequência);

$W_{Z_{s,d}}$ = peso do menor caminho de *backup* entre s e d ;

$\Omega(P_n, Z, r(s, d, b))$: função que procura um caminho de *backup* em Z em que P_{Z_n} são caminhos disjuntos de P_n e satisfazem a requisição b_m ;

$\Phi(\tilde{G}_{n,b_m}, P_n, r(s, d, b))$: função que produz um menor caminho de *backup* Z_n entre s and d em \tilde{G}_{n,b_m} , em que Z_n e P_{Z_n} são caminhos disjuntos para P_n , e que o caminho satisfaz a requisição b_m ;

T_n : p -cycle contendo os vértices u e v e arestas correspondendo ao mapeamento de b_m arestas do multigrafo G ;

T : conjunto de todos os p -cycles ativos;

P_{T_n} : conjunto de todos os caminhos protegidos pelo p -cycle T_n ;

$W(T_n)$: o peso do p -cycle T_n (a soma dos pesos de todas as arestas);

$W_{T_{s,d}}$ = peso do p -cycle que protegerá o caminho entre s e d ;

$v(P_n, T, r(s, d, b))$: função que procura um p -cycle em T que P_{T_n} são caminhos disjuntos de P_n e satisfazem a requisição b_m e que s e d fazem parte do p -cycle;

$\theta(\tilde{G}_{n,b_m}, P_n, r(s, d, b))$: função que produz um menor p -cycle T_n , entre s and d em \tilde{G}_{n,b_m} , que P_{T_n} são caminhos disjuntos para P_n , e que o p -cycle satisfaz a requisição b_m ;

$\Psi(\tilde{G}_{n,b}, P_n, r(s, d, b))$: função que produz um p -cycle T_n , entre s and d em $\tilde{G}_{n,b}$, que satisfaz a requisição b_m , e é disjunto de P_n (p -cycle é transzonal a P_n) e que P_{T_n} são caminhos disjuntos para P_n ;

4.3 O algoritmo BARTMAN

O algoritmo *BAckuping, Routing, specTrum, coRe and Modulation level AssigNment* (BARTMAN) procura o formato de modulação apropriado para um determinado pedido, a partir dos níveis mais altos de modulação. Os formatos de modulação com níveis mais elevados têm uma maior eficiência espectral que os formatos de modulação com níveis mais baixos. Por outro lado, a gama de transmissão de formatos de modulação de alto nível é menor que o alcance dos formatos de modulação de baixo nível. O formato de modulação é selecionado de acordo a manter a qualidade da transmissão (QoT).

Como mencionado no início deste capítulo, o algoritmo BARTMAN (algoritmo 5) é uma extensão do algoritmo 2. Este algoritmo segue a mesma estrutura dos algoritmos anteriores, escolhendo primeiramente o caminho primário, e em seguida o caminho de backup. No entanto, a diferença fundamental entre o algoritmo SBPPMC e o algoritmo BARTMAN é a inserção da escolha de diferentes formatos de modulação na criação dos caminhos. No algoritmo 5, a complexidade de transformar multígrafos em grafos aumenta, visto que, conforme a modulação é alterada, há a necessidade de criação de novos grafos. Para uma dada requisição, o algoritmo escolhe a modulação com a maior eficiência espectral e transforma o multigrafo em $C \times (N - b_m + 1)$ grafos (linha 2), a criação dos grafos depende exclusivamente da banda requerida e da capacidade do *slot* utilizando determinado formato de modulação. Com os grafos gerados, o algoritmo de Dijkstra é utilizado para encontrar o menor caminho (linha 3). A linha 4 escolhe o menor caminho e o peso do caminho dentre os menores caminhos dos grafos. Na linha 6, as arestas do caminho escolhido têm seus pesos configurados para ∞ . A linha 1 escolhe o melhor formato de modulação para gerar o menor caminho. Caso não seja possível encontrar um menor caminho (linha 7), a requisição é bloqueada (linha 8). Caso contrário um caminho de *backup* compartilhado para proteger o caminho primário deverá ser buscado dentre os caminhos de *backup* ativos (linha 10). Um determinado caminho de *backup* só poderá proteger caminhos primários disjuntos. Se houver um caminho que possa ser compartilhado para proteger a requisição, a requisição é aceita (linha 11). Se não houver nenhum caminho que possa proteger o caminho primário, então um caminho de *backup* precisa ser criado. A criação do caminho de

Algoritmo 5 BARTMAN

Entrada: $G, r(s, d, b)$
Saída: Caminho primário e Caminho de backup

```

1: Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
2:    $\tau(G, C, b_m)$ 
3:    $(W(P_n), P_n) = \alpha(\tilde{G}_{n,b_m}, r(s, d, b)) \quad \forall n \in \sigma$ 
4:    $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
5: Fim Para todo
6:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
7: Se  $W_{P_{s,d}} = \infty$  então
8:   bloqueia requisição  $(r(s, d, b))$ 
9: Senão
10:  Se  $\exists \Omega(P_n, Z, r(s, d, b))$  então
11:    estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
12:  Senão
13:    Para todo  $m \in M \wedge W_{Z_{s,d}} = \infty$  faça
14:       $\tau(G, C, b_m)$ 
15:       $(W(Z_n), Z_n) = \Phi(\tilde{G}_{n,b_m}, P_n, r(s, d, b)) \quad \forall n$ 
16:       $W_{Z_{s,d}} = W(Z_n) \mid \forall i W(Z_n) \leq W(Z_i)$ 
17:    Fim Para todo
18:    Se  $W_{Z_{s,d}} = \infty$  então
19:      bloqueia requisição  $(r(s, d, b))$ 
20:    Senão
21:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
22:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in Z_i$ 
23:    Fim Se
24:  Fim Se
25: Fim Se

```

backup pode utilizar formato de modulação diferente do caminho primário. Logo, o processo de transformação do multigrafo em grafos necessita ser refeito (linhas 13 e 14). As linhas 15 e 16 buscam o menor caminho de *backup* disjunto do caminho primário para o formato de modulação. Caso nenhum caminho de *backup* possa ser criado, a requisição é bloqueada (linha 19). Caso contrário, a requisição é aceita, os caminhos são estabelecidos e as correspondentes arestas do multigrafo G tem seus pesos alterados para ∞ .

A complexidade do algoritmo BARTMAN é analisada como segue. No pior caso, a modulação escolhida será a que tem menor capacidade, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $M \times (E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b_m)$ grafos, onde a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, no pior caso para o caminho de backup, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b_m)$ grafos. Logo, no pior caso, o algoritmo BARTMAN tem complexidade de $2 \times (M \times (E + V) + C \times (N - b_m) \times (E + V \log V))$. Como N , C e b_m podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

4.4 O algoritmo PERFECTA

Nesta seção, propõe-se um algoritmo chamado *Protecting, Routing, modulation Format, core and spectrum Allocation* (PERFECTA), o algoritmo provê proteção para redes ópticas elásticas utilizando multiplexação espacial através de *p-cycles* FIPP, neste algoritmo os *p-cycles* bem como os caminhos primários das requisições são criados escolhendo-se os formatos de modulação e através do processo de transformação do multigrafo em grafos. Como no algoritmo 5, no algoritmo 6, o formato de modulação é selecionado de forma a manter a qualidade da transmissão (QoT). O algoritmo PERFECTA é uma extensão do algoritmo FIPPMC. A diferença entre o algoritmo 3 (FIPPMC) e o algoritmo 6 (PERFECTA) é a inserção da escolha de diferentes formatos de modulação na criação dos caminhos.

Algoritmo 6 PERFECTA

Entrada: $G, r(s, d, b)$

Saída: Caminho primário e Caminho de backup

- 1: **Para todo** $m \in M \wedge W_{P_{s,d}} = \infty$ **faça**
 - 2: $\tau(G, C, b_m)$
 - 3: $(W(P_n), P_n) = \alpha(\tilde{G}_{n,b_m}, r(s, d, b)) \quad \forall n \in \sigma$
 - 4: $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$
 - 5: **Fim Para todo**
 - 6: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$
 - 7: **Se** $W_{P_{s,d}} = \infty$ **então**
 - 8: bloqueia requisição ($r(s, d, b)$)
 - 9: **Senão**
 - 10: **Se** $\exists v(P_n, T, r(s, d, b))$ **então**
 - 11: estabelece requisição ($r(s, d, b)$) **como** P_n **e** T_n
 - 12: **Senão**
 - 13: **Para todo** $m \in M \wedge W_{T_{s,d}} = \infty$ **faça**
 - 14: $\tau(G, C, b_m)$
 - 15: $(W(T_n), T_n) = \theta(\tilde{G}_{n,b_m}, P_n, r(s, d, b)) \quad \forall n$
 - 16: $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$
 - 17: **Fim Para todo**
 - 18: **Se** $W_{T_{s,d}} = \infty$ **então**
 - 19: bloqueia requisição ($r(s, d, b)$)
 - 20: **Senão**
 - 21: estabelece requisição ($r(s, d, b)$) **como** P_n **e** T_n
 - 22: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$
 - 23: **Fim Se**
 - 24: **Fim Se**
 - 25: **Fim Se**
-

Para o algoritmo PERFECTA a linha 1 escolhe o formato de modulação a ser utilizado pelo menor caminho. A linha 2 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação. As linhas 3 e 4 computam o menor caminho e o custo a partir dos grafos gerados. O processo de escolha do formato de modulação (linha 1) é repetido até que um caminho seja encontrado ou não exista mais formato de modulação. Na linha 6, o peso das arestas do caminho escolhido é marcado com ∞ . Se nenhum caminho primário for

encontrado (linha 7), a requisição é bloqueada (linha 8). Na linha 10, busca-se um *p-cycle* dentre os *p-cycles* ativos que possa proteger a requisição. Para a escolha do *p-cycle*, é levado em consideração que todos os caminhos primários protegidos por um *p-cycle* devem ser disjuntos, os nós finais do caminho primário precisam estar no *p-cycle*, e além disso, o possível caminho a ser utilizado pela requisição precisa respeitar a eficiência espectral. Com a utilização de *p-cycle* e modulação, deve-se levar em conta a distância máxima do caminho utilizado e não a distância de percorrer todo o *p-cycle*. Caso o caminho primário estiver no *p-cycle*, a distância a ser levada em consideração é a dos enlaces que não fazem parte do caminho. Por outro lado, para o caminho primário que é transzonal ao *p-cycle*, a distância considerada será a menor distância entre os nós de origem e destino. Caso exista um *p-cycle* que possa proteger a requisição, a conexão é estabelecida (linha 10). Caso contrário, um *p-cycle* para proteger o caminho primário tem que ser criado. Para criação do *p-cycle*, é escolhido um formato de modulação (linha 13) e gerado $C \times (N - b_m + 1)$ grafos a partir do multigrafo (linha 14). As linhas 15 e 16 são as responsáveis pela criação do *p-cycle* com menor custo. Se o *p-cycle* não puder ser criado, a requisição é bloqueada (linha 19). Caso contrário, a requisição é aceita e o *p-cycle*, bem como o caminho primário, são estabelecidos (linha 21) e as arestas correspondentes são configuradas para ∞ (linhas 22).

A complexidade do algoritmo PERFECTA é analisada da seguinte forma: No pior caso a modulação escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $M \times (E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b_m)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para o *p-cycle*, o algoritmo de Suurballe [5] é executado M vezes em $C \times (N - b_m)$ grafos, em que a complexidade de Suurballe é $O(E + V \log V)$. Com isso, o algoritmo PERFECTA tem complexidade de $M \times (E + V) + M \times (C \times (N - b_m) \times (E + V \log V)) + M \times (C \times (N - b_m) \times (E + V \log V))$. Como N , C e b_m podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

4.5 O Algoritmo MMIFMC

O algoritmo introduzido nesta seção, chamado *Modulation, Minimum Interference and Failure-independent path protecting for MultiCore networks* (MMIFMC) estende o algoritmo 4 (MIFMC), para realizar proteção dos caminhos ópticos através do uso de *p-cycles* FIPP, interferência mínima e modulação. O algoritmo MMIFMC estende o MIFMC, pela introdução da escolha do formato de modulação tanto no caminho de *backup* quanto para o *p-cycle*. O algoritmo MMIFMC garante proteção para todos os caminhos primários. A reserva de recursos para criar o *p-cycle* FIPP pode sobrecarregar os enlaces na rede, uma vez que o *p-cycle* que protege os caminhos no ciclo usam os mesmos enlaces do caminho primário. Por outro lado, o *p-cycle* que protege caminhos transzonais tende a reservar mais recursos, pois tem um maior número de saltos. Os *p-cycles* que protegem caminhos primários no ciclo utilizam menos recursos da rede do que *p-cycles* que protegem caminhos transzonais. Por outro lado, eles podem sobrecarregar os enlaces ao longo de um caminho. Por conseguinte, é necessário adotar critérios para evitar a formação de gargalos, equilibrando a carga entre potenciais caminhos; é necessário adotar uma

abordagem de interferência mínima para evitar o bloqueio de requisições.

Algoritmo 7 MMIFMC

Entrada: $G, r(s, d, b)$

Saída: Caminho primário e Caminho de backup

```

1: Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
2:    $\tau(G, C, b_m)$ 
3:    $(W(P_n), P_n) = \alpha(\tilde{G}_{n,b_m}, r(s, d, b)) \quad \forall n \in \sigma$ 
4:    $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
5: Fim Para todo
6:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
7: Se  $W_{P_{s,d}} = \infty$  então
8:   bloqueia requisição  $(r(s, d, b))$ 
9: Senão
10:  Se  $\exists v(P_n, T, r(s, d, b))$  então
11:    estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
12:  Senão
13:    Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
14:       $\tau(G, C, b_m)$ 
15:       $(W(T_n), T_n) = \Psi(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
16:       $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
17:    Fim Para todo
18:    Se  $W_{T_{s,d}} = \infty$  então
19:      Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
20:         $\tau(G, C, b_m)$ 
21:         $(W(T_n), T_n) = \theta(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
22:         $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
23:      Fim Para todo
24:      Se  $W_{T_{s,d}} = \infty$  então
25:        bloqueia requisição  $(r(s, d, b))$ 
26:      Fim Se
27:    Fim Se
28:    Se  $W_{T_{s,d}} \neq \infty$  então
29:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
30:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
31:    Fim Se
32:  Fim Se
33: Fim Se

```

No algoritmo MMIFMC, a linha 1 estabelece o conjunto de modulações que será testado na criação do caminho primário. A linha 2 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação. As linhas 3 e 4 computam o caminho com o menor custo a partir dos grafos gerados. O processo de escolha do formato de modulação (linha 1) é repetido até que um caminho seja encontrado ou não exista mais formato de modulação a ser testados. Se o peso do caminho mais curto é ∞ (linha 7), não foi possível encontrar um caminho sob a restrição de contiguidade para a demanda b com a alocação começando com o n -ésimo *slot*; portanto, a requisição é bloqueada (Linha 8). Caso contrário, um p -cycle para

proteger o caminho óptico é pesquisado dentre os p -cycles ativos que já estão protegendo outras requisições (linha 10). Os caminhos que já estão sendo protegidos pelo p -cycle necessitam ser disjuntos do novo caminho primário já selecionado. Além disso, os nós finais do caminho primário precisam estar no p -cycle. Caso exista um p -cycle, o caminho óptico é estabelecido (linha 11). Caso contrário, um p -cycle com interferência mínima para proteger o caminho óptico deve ser criado. A linha 13 escolhe um formato de modulação pra o p -cycle que será criado. A linha 14 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação. Para evitar gargalos na rede e adotando o conceito de interferência mínima, o algoritmo tenta criar um p -cycle transzonal ao caminho (Linha 15). Se o p -cycle transzonal for criado, o p -cycle e o caminho são estabelecidos para satisfazer a requisição (linha 29) e as arestas correspondentes no multigrafo G tenham o seu peso alterado para ∞ (Linha 30 e 31), o que significa que os *slots* foram alocados para o caminho óptico recém-criado. Caso nenhum p -cycle transzonal possa ser criado para proteger o caminho óptico, então o algoritmo tenta criar um menor p -cycle. A linha 19 escolhe um formato de modulação pra o p -cycle que será criado. A linha 20 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação. Caso nenhum p -cycle possa ser criado a requisição é bloqueada (Linha 25). Se o p -cycle for criado, o p -cycle e o caminho são estabelecidos para satisfazer o pedido (linha 29) e as arestas correspondentes terão o seu peso alterado para ∞ (Linha 30), o que significa que os *slots* foram alocados para o caminho óptico recém-criado.

A complexidade do algoritmo MMIFMC é analisada como segue: No pior caso a modulação escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $O(E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b)$ grafos, onde a complexidade amortizada de Dijkstra é $O(E + V \log V)$. No pior caso, o algoritmo não encontrará um caminho transzonal, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $O(E + V)$ e executando o algoritmo de Suurballe [5] M vezes em $C \times (N - b)$ grafos e logo após no pior caso a modulação escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $O(E + V)$ e executando o algoritmo de Suurballe mais M vezes, onde a complexidade de Suurballe é $O(E + V \log V)$. Com isso, o algoritmo MMIFMC tem complexidade de $M \times (E + V) + M \times (C \times (N - b_m) \times (E + V \log V)) + M \times (C \times (N - b_m) \times (E + V \log V)) + M \times (C \times (N - b_m) \times (E + V \log V))$. Como N , C e b_m podem ser expressados como constantes a complexidade do algoritmo é $O(E + V \log V)$.

4.6 Resultados numéricos

Nas figuras, as curvas rotuladas como PERFECTA mostram os resultados para as redes que utilizam o algoritmo PERFECTA (Algoritmo 6), as curvas rotuladas como MMIFMC mostram os resultados para as redes que utilizam o algoritmo MMIFMC (Algoritmo 7), e as curvas rotuladas como BARTRMAN mostram os resultados para as redes que utilizam o algoritmo BARTRMAN (Algoritmo 5), as curvas rotuladas como SSCAM encontram o caminho primário através do algoritmo proposto em [102], utilizando $k = 3$ e uma escolha de modulação baseada no tamanho do caminho. O caminho de *backup*, mantém as mesmas propriedades e adiciona o

compartilhamento de caminhos de backup. As curvas rotuladas como Cap-DPPM mostram os resultados para as redes que utilizam o algoritmo proposto em [100] e utiliza uma escolha de modulação baseada no tamanho do caminho.

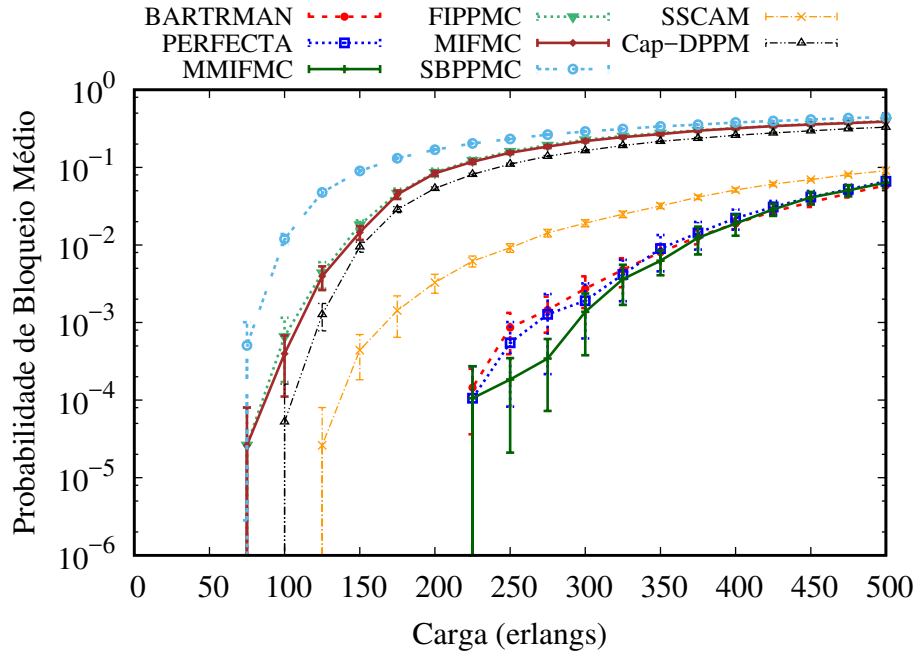


Figura 4.2: Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia Pan-European.

A Figura 4.2 mostra a probabilidade de bloqueio para a topologia Pan-European. Os algoritmos Cap-DPPM e SSCAM iniciam o bloqueio de requisições sob cargas de 100 e 125 erlangs, respectivamente. Os algoritmos BARTMAN, PERFECTA e MMIFMC iniciam o bloqueio de requisições sob cargas de 200 erlangs. Sob todas as cargas simuladas o algoritmo BARTMAN produz valores de BBR menores do que os valores de BBR produzidos pelo algoritmo SSCAM, apesar de proverem proteção de caminho de *backup* compartilhado. O algoritmo MMIFMC que utiliza multigrafo, modulação, *p-cycle* FIPP e interferência mínima na criação do *p-cycle*, produz as BBR mais baixas para todas as cargas simuladas na topologia apresentada. Os algoritmos SBPPMC, FIPPMC e MIFMC, que não utilizam modulação adaptativa, produzem os maiores valores de BBR para todas as cargas simulada. Isto evidencia as vantagens de utilizar modulação adaptativa quando comparado com os algoritmos BARTMAN, PERFECTA e MMIFMC. Para todas as cargas simuladas, o algoritmo MMIFMC produz valores de BBR menor do que o algoritmo PERFECTA, pois, com a maior conectividade da topologia, a criação de *p-cycles* transzonais ao caminhos são mais viáveis. O algoritmo BARTMAN produz resultados próximos aos resultados dos algoritmos que utilizam *p-cycle*, isto ocorre pois apesar do algoritmo BARTMAN utilizar menos recursos para proteção individual, ele produz um menor compartilhamento.

A Figura 4.3 mostra a relação de *crosstalk* por *slot* gerado para a topologia Pan-European. Os valores de CpS gerados para os algoritmos FIPPMC e MIFMC iniciam com valores de 0,018 e 0,021 e aumentam até 0,39 e 0,41, respectivamente. Os valores de CpS gerados para os al-

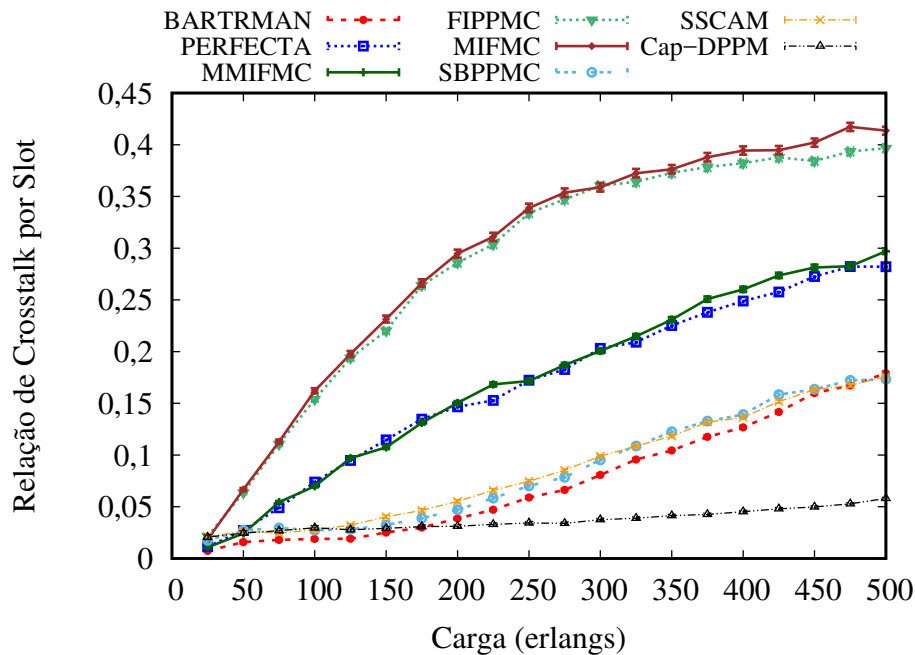


Figura 4.3: *Crosstalk* por *slot* dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia Pan-European.

algoritmos PERFECTA e MMIFMC iniciam com um valor de 0,1 e aumentam até 0,28 e 0,29, respectivamente. Sob cargas altas, o algoritmo Cap-DPPM produz o menor CpS. O algoritmo Cap-DPPM, inicia com um valor de 0,02 e aumenta até 0,06. O algoritmo SSCAM gera valores de CpS iniciando em 0,02 e cresce até 0,18. Sob cargas baixas, o algoritmo BARTMAN produz o menor CpS, o algoritmo BARTMAN inicia CpS com um valor de 0,007 e aumenta até 0,18. O algoritmo BARTMAN produz um CpS próximo ao produzido pelo algoritmo SSCAM apesar do BARTMAN produzir um baixo valor de BBR quando comparado aos algoritmos SSCAM e Cap-DPPM. Os algoritmos BARTMAN, Cap-DPPM e SSCAM produzem um menor CpS pois os algoritmos PERFECTA e MMIFMC possuem um maior número de saltos do caminho de *backup* exigindo mais recursos da rede na criação do p-cycle. A relação de CpS identifica apenas os *slots* que sofrem com crosstalk, no entanto o *crosstalk* pode ser ainda aceitável. Os algoritmos SBPPMC, FIPPMC e MIFMC, que não utilizam modulação adaptativa, apresenta um maior CpS do que os algoritmos BARTMAN, PERFECTA e MMIFMC, que utilizam modulação. Isto acontece pois os algoritmos BARTMAN, PERFECTA e MMIFMC aceitam um maior número de requisições.

A Figura 4.4 mostra a razão de fragmentação gerada para a topologia Pan-European. Até cargas de 275 erlangs, o algoritmo BARTMAN produz a menor razão de fragmentação. Sob cargas menores que 225 erlangs o algoritmo BARTMAN não produz BBR e além disso produz uma razão de fragmentação baixa. A partir de cargas de 325 erlangs, o algoritmo Cap-DPPM produz a menor razão de fragmentação, uma vez que ele aceita um menor número de requisições, gerando menos fragmentação. Sob baixas cargas, os algoritmos SBPPMC, FIPPMC e MIFMC, que não utilizam modulação adaptativa, produzem as maiores razões de fragmentação. Isto ocorre pois os outros algoritmos que utilizam modulação adaptativa podem ajustar o

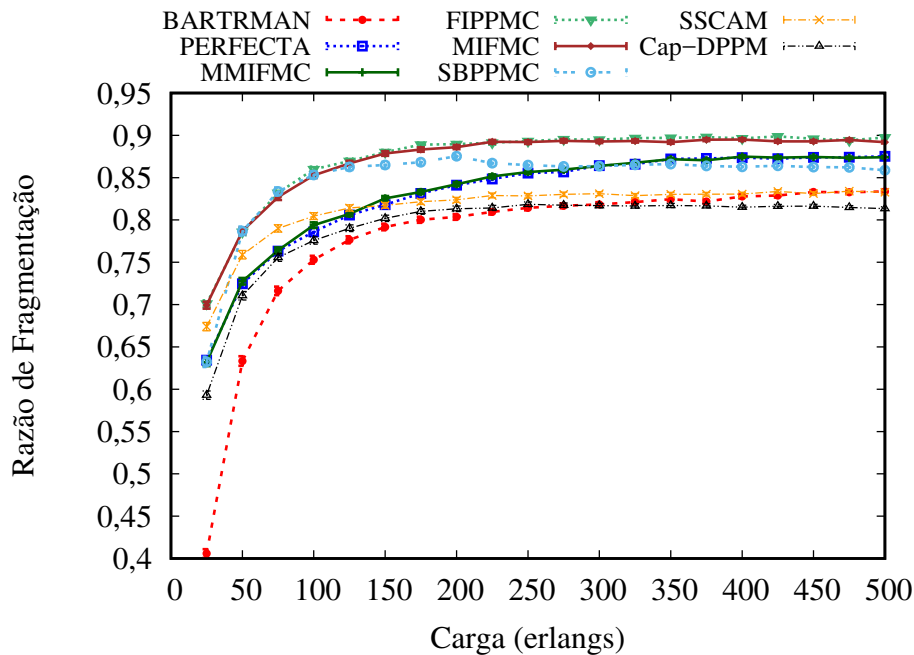


Figura 4.4: Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia Pan-European.

formato de modulação de acordo com os requisitos de cada demanda, possibilitando uma alocação de recursos melhor distribuída. Sob cargas maiores que 275 erlangs, o algoritmo SBPPMC produz razão de fragmentação menor que os algoritmos PERFECTA e MMIFMC. Isto acontece pois os *p-cycles* alocados pelos algoritmos PERFECTA e MMIFMC alocam muitos recursos o que gera maior razão de fragmentação. Sob cargas maiores que 150 erlangs, os algoritmos PERFECTA e MMIFMC produzem a maior razão de fragmentação devido ao maior número de saltos do caminho de backup, o que gera uma maior fragmentação ao realizar o estabelecimento das conexões. A diferença na razão de fragmentação dos algoritmos Cap-DPPM e SSCAM ocorre devido ao maior número de requisições aceita pelo algoritmo SSCAM, o que gera um maior estabelecimento das conexões e, conseqüentemente, maior fragmentação.

A Figura 4.5 mostra a eficiência energética gerada para a topologia Pan-European. O algoritmo Cap-DPPM produz uma maior eficiência energética para todas as cargas simuladas, que está relacionada a maior BBR gerada. A maior eficiência energética do algoritmo Cap-DPPM aumenta conforme aumenta o bloqueio e conseqüentemente diminui o número de saltos. Sob cargas mais baixas, o algoritmo BARTMAN produz maior eficiência energética que os algoritmos PERFECTA, MMIFMC, SSCAM. Isto acontece pois este algoritmo produz um menor número de saltos do caminho primário. Os algoritmos SBPPM, FIPPMC e MIFMC produzem eficiência energética semelhantes. Além disso, estes algoritmos produzem os menores valores de eficiência energética, pois a utilização de modulação fixa, subutiliza o espectro realizando maior consumo energético.

A Figura 4.6 mostra o número médio de saltos do caminho primário gerado para a topologia Pan-European. A utilização de diferentes formatos de modulação e do multigrafo pelos algoritmos BARTMAN, PERFECTA e MMIFMC para encontrar o menor caminho primário,

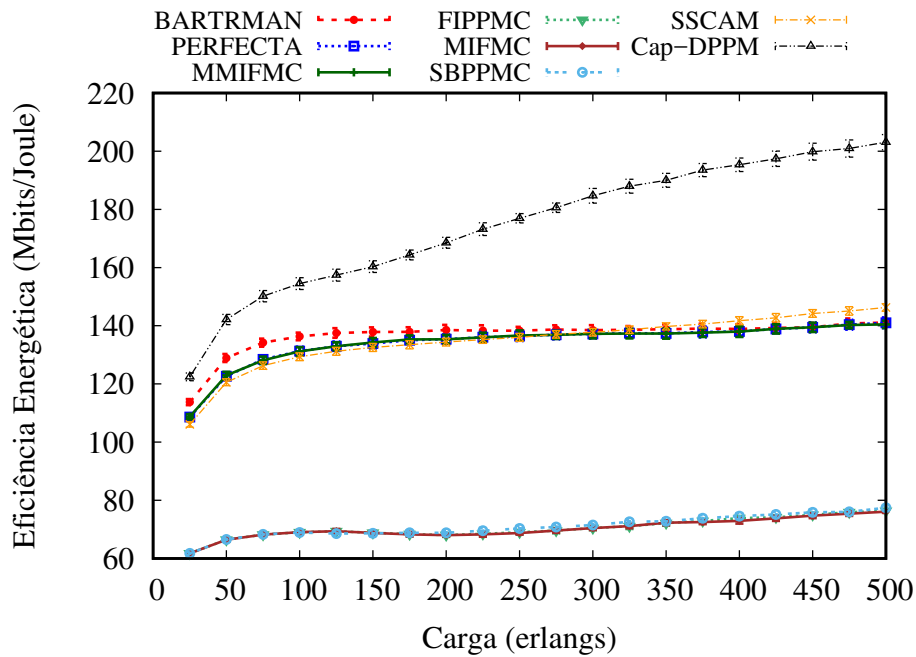


Figura 4.5: Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia Pan-European.

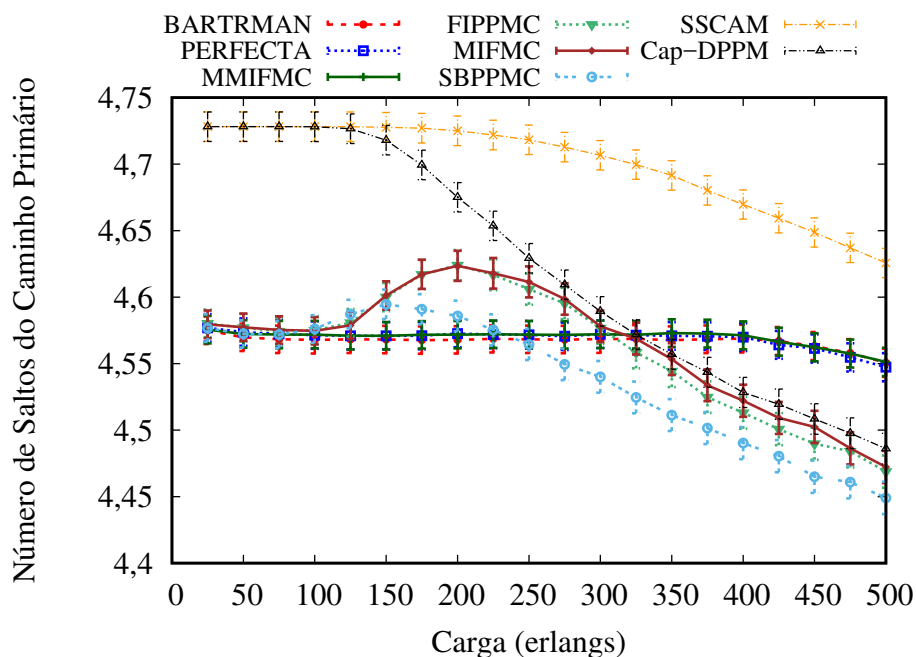


Figura 4.6: Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia Pan-European.

principalmente para cargas mais baixas, produz um menor número de saltos quando comparado aos algoritmos Cap-DPPM e SSCAM que encontram caminhos mais curtos para posteriormente encontrar *slots* disponíveis obedecendo contiguidade e continuidade do espectro. Por outro lado, sob cargas altas os algoritmos FIPPMC, MIFMC e SBPPMC produzem caminhos

primários com número de saltos menores. Isto acontece, pois os algoritmos FIPPMC, MIFMC e SBPPMC produzem valores de BBR maiores, o que ocasiona que apenas menores caminhos são alocados. Os algoritmos PERFECTA e MMIFMC que usam p -cycle produzem uma maior número de saltos que o algoritmo BARTRMAN que utiliza caminhos de *backup* compartilhados para todas as cargas testadas. No entanto, conforme esses algoritmos produzem valores de BBR mais próximos, mais próximos também é o número de saltos do caminho primário, evidenciando a influencia da BBR com relação ao número de saltos do caminho primário para os algoritmos que utilizam multigrafo.

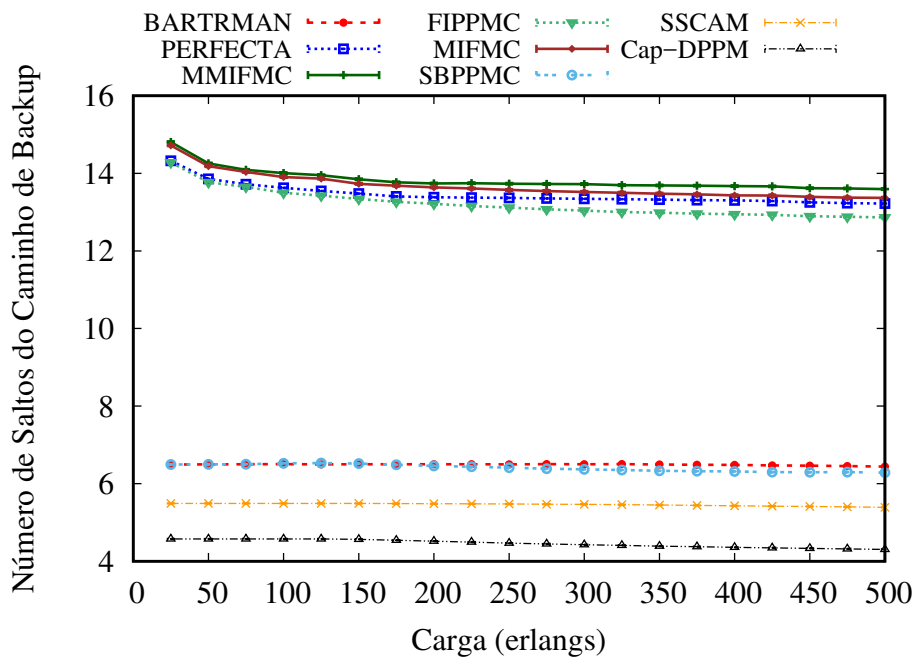


Figura 4.7: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia Pan-European.

A Figura 4.7 mostra o número médio de saltos do caminho de *backup* gerado para a topologia Pan-European. Os algoritmos que utilizam p -cycle produzem caminhos de *backup* com quase três vezes o número de saltos do que os que não utilizam p -cycle. O algoritmo MMIFMC produz caminhos de *backup* com número médio de saltos sempre maior que o algoritmo PERFECTA, isto ocorre pois o algoritmo MMIFMC prioriza p -cycles transzonais ao caminho primário, e esses p -cycles tendem a ser maiores que os caminhos que usam os mesmos enlaces do p -cycle. Os algoritmos BARTRMAN, PERFECTA e MMIFMC que utilizam modulação adaptativa produziram sempre maior número de saltos do que os algoritmos SBPPMC, FIPPMC e MIFMC que utilizam modulação fixa. O algoritmo BARTRMAN produz caminhos de *backup* com número médio de saltos uma unidade maior que o SSCAM que por sua vez produz caminhos de *backup* com número médio de saltos uma unidade maior que o Cap-DPPM, isto ocorre devido a forma de criação dos caminhos de *backup*.

A Figura 4.8 mostra a probabilidade de bloqueio para a topologia USA. Enquanto o algoritmo Cap-DPPM inicia o bloqueio de requisições sob cargas de 100 erlangs, SSCAM inicia o bloqueio somente sob cargas de 125 erlangs. Enquanto os algoritmos FIPPMC e MIFMC

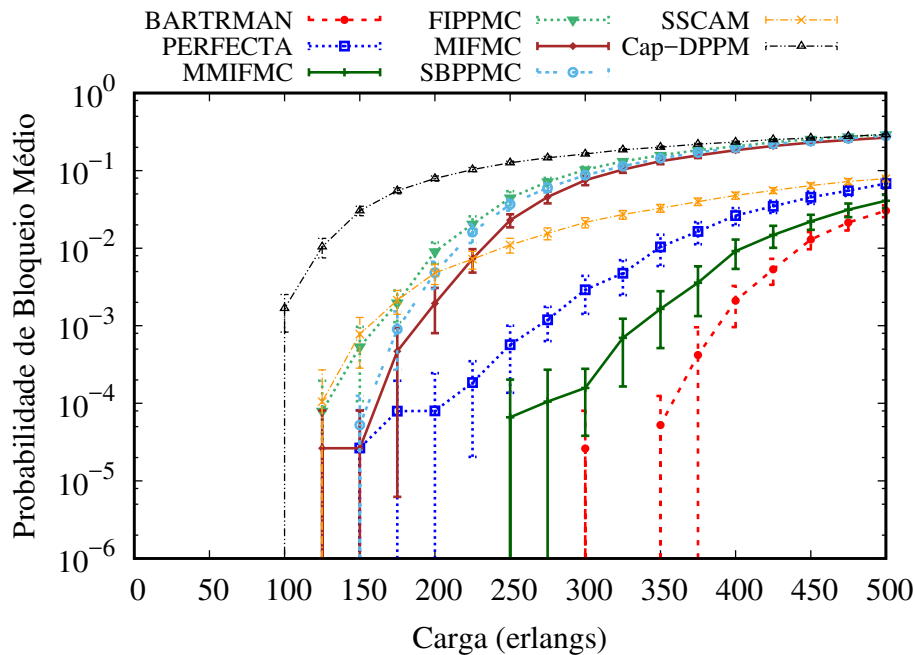


Figura 4.8: Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.

iniciam o bloqueio de requisições sob cargas de 125 erlangs, o algoritmo SBPPMC inicia o bloqueio somente sob cargas de 150 erlangs. Por sua vez, os algoritmos PERFECTA e MMIFMC iniciam o bloqueio de requisições a partir de 150 e 250 erlangs, respectivamente. E o algoritmo BARTRMAN produz valores de BBR apenas sob carga de 300 erlangs. Nesta topologia, os algoritmos produzem BBRs visivelmente distintas. O algoritmo BARTRMAN produz valores de BBR menores que os todos os algoritmos simulados. Isto acontece pois os nós da topologia USA, com exceção de alguns nós de bordas, possuem pelo menos outros três nós conectados a eles. Esta característica da topologia, juntamente com o uso do multigrafo, e de diferentes formatos de modulação e do uso do algoritmo Dijkstra, possibilita os menores valores de BBR entre os algoritmos apresentados. O maiores valores de BBR produzido pelo algoritmo Cap-DPPM devido ao fato de realizar proteção 1:1, o que gera reserva de recursos individuais para todos os caminhos estabelecidos, e, conseqüentemente, impossibilita o estabelecimento de algumas conexões. Os valores de BBR gerados pelo algoritmo SSCAM, por sua vez, diferencia-se dos valores de BBR produzidos pelos algoritmos PERFECTA, BARTRMAN e MMIFMC pela utilização do multigrafo que distribui a alocação de recursos. Os algoritmos SBPPM, FIPPMC e MIFMC apesar de não utilizarem modulação adaptativa produzem BBR sob cargas mais altas do que as produzidas pelos algoritmos SSCAM e Cap-DPPM. Os algoritmos PERFECTA e MMIFMC apresentam resultados visivelmente distintos, evidenciando ainda mais o benefício da utilização do conceito de interferência mínima na criação do *p-cycle* FIPP.

A Figura 4.9 mostra a relação de *crossstalk* por *slot* gerado para a topologia USA. Os valores de CpS gerados pelos os algoritmos BARTRMAN e SSCAM iniciam, respectivamente, em 0,02 e 0,03 e aumentam até 0,18. Os valores de CpS gerados pelos algoritmos PERFECTA e MMIFMC iniciam em de 0,02 e aumentam até 0,28 e 0,30, respectivamente. Os valores de CpS gerados pelo algoritmo Cap-DPPM iniciam em 0,03 e permanecem quase constante aumen-

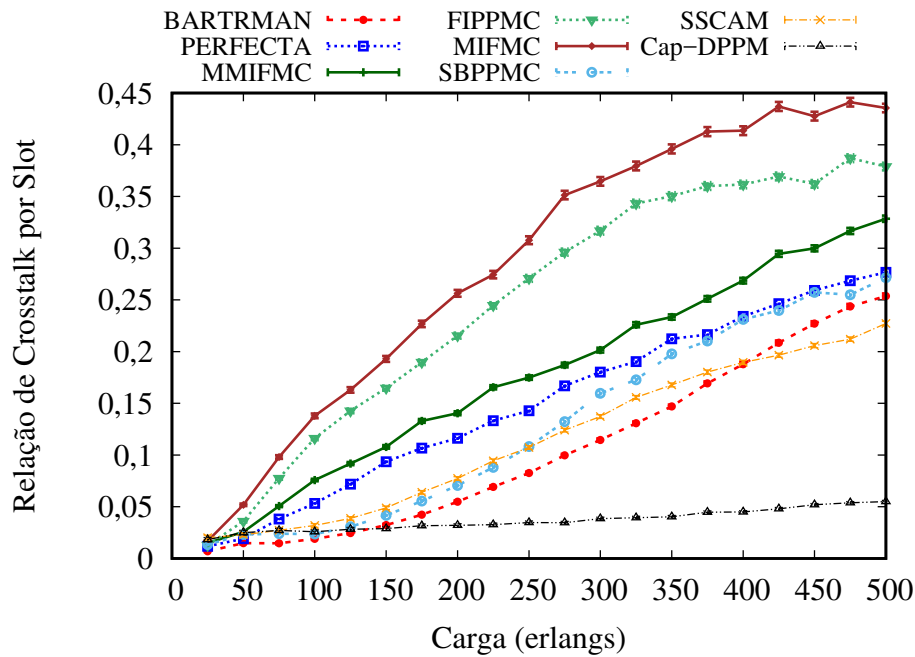


Figura 4.9: *crosstalk* por *slot* dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia USA.

tando apenas até o valor de 0,06. Os valores de CpS gerados pelos algoritmos MIFMC, FIPPMC e SBPPMC iniciam em de 0,02 e aumentam, respectivamente, até 0,48, 0,38 e 0,28, respectivamente. Os valores de CpS produzidos pelos algoritmos FIPPMC e MIFMC que usam *p-cycle* e modulação fixa são superiores aos produzidos pelos algoritmos PERFECTA e MMIFMC. Por sua vez, os algoritmos PERFECTA e MMIFMC produzem valores de CpS maiores que outros algoritmos. Isto acontece como consequência da maior utilização de recursos, uma vez que a criação de *p-cycles* necessita de um maior perímetro, fazendo com que mais *slots* sejam afetados pelo *crosstalk*. Além do algoritmo BARTMAN produzir valores de BBR baixo e uma alta utilização, ele também produz valores CpS inferiores aos produzidos pelos algoritmos PERFECTA, MMIFMC e SSCAM, como consequência do uso intercalado de núcleos para os menores caminhos primários e dos menores caminhos de backup, diminuindo, assim, os valores de CpS gerados.

A Figura 4.10 mostra a razão de fragmentação gerada para a topologia USA. Sob cargas de até 375 erlangs o algoritmo BARTMAN produz uma razão de fragmentação menor que a dada pelo outros algoritmo simulados, principalmente para cargas mais baixas. Isto contribui para geração de menor BBR por este algoritmo. Os algoritmos FIPPMC e MIFMC, que utilizam modulação fixa e *p-cycle* FIPP produzem a maior razão de fragmentação. Apesar do algoritmo SBPPMC utilizar modulação fixa, sob baixas cargas este algoritmo produz baixa razão de fragmentação. Os algoritmos PERFECTA e MMIFMC geram uma fragmentação maior que o algoritmo BARTMAN o que contribui para um maior bloqueio de requisições, visto que uma fragmentação mal realizada prejudica a utilização dos recursos da rede por novos caminhos a serem criados. Como os algoritmos SSCAM e Cap-DPPM encontram a rota sem verificar a disponibilidade de *slots* e núcleos para criação do caminho primário, eles possuem uma grande semelhança entre a razão de fragmentação, a diferença ocorre devido o compartilhamento utili-

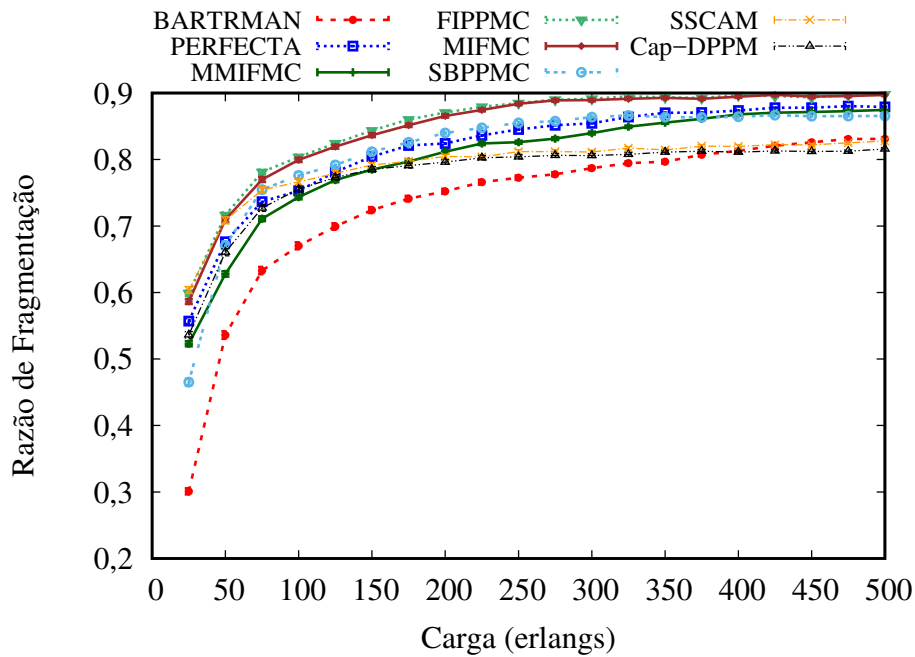


Figura 4.10: Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.

zado pelo algoritmo SSCAM.

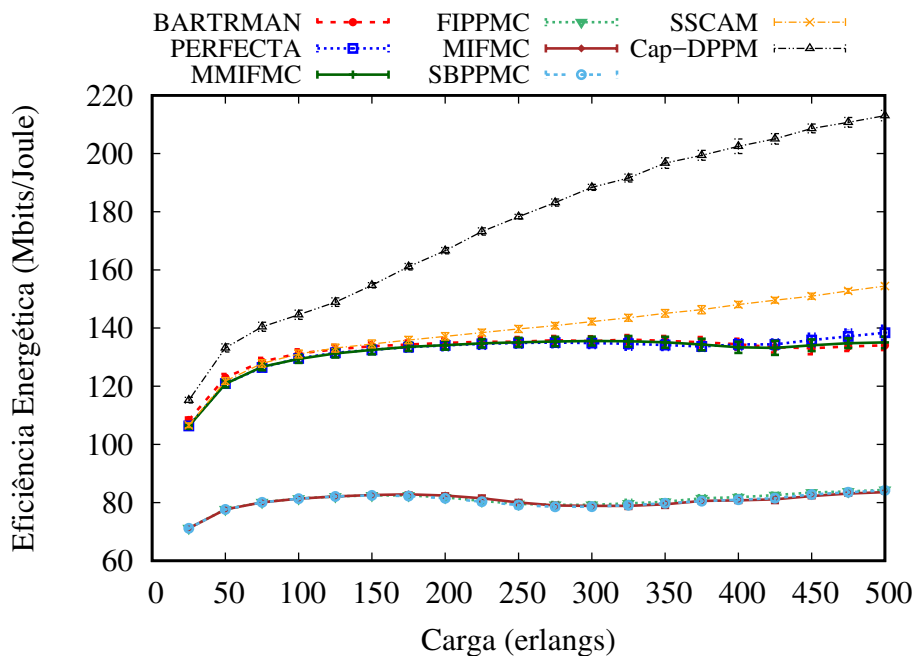


Figura 4.11: Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.

A Figura 4.11 mostra a eficiência energética gerada para a topologia USA. O algoritmo Cap-DPPM produz maior eficiência energética que os outros algoritmos simulados, o que é

consequência de encontrar rotas baseadas apenas em caminhos curtos. Além disso, por produzir um alto BBR, os algoritmos que possuem caminhos mais longos tendem a não ser estabelecidos. Sob cargas baixas, não há muita diferença entre a eficiência energética do SSCAM e os algoritmos que utilizam multigrafo. A diferença de eficiência energética surge apenas para cargas pesadas. Para estas cargas, os algoritmos que utilizam o multigrafo produzem valores de BBR significativamente mais baixos. Os algoritmos SBPPMC, FIPPMC e MIFMC produzem os menores valores de eficiência energética quando comparado aos outros algoritmos. Isto acontece devido ao maior consumo energético gerado quando utilizado modulação fixa.

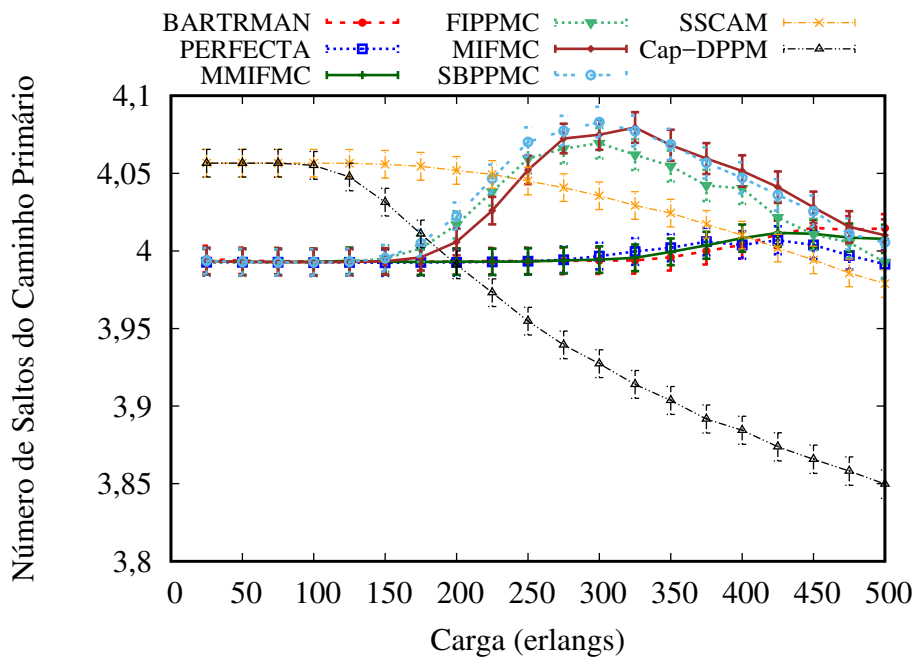


Figura 4.12: Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia USA.

A Figura 4.12 mostra o número médio de saltos do caminho primário gerado para a topologia USA. Sob baixas cargas, os algoritmos BARTMAN, PERFECTA, MMIFMC, SBPPMC, FIPPMC e MIFMC, que utilizam o multigrafo, possuem número médio de saltos do caminho primário semelhantes para qualquer carga simulada, independente da utilização de modulação. Apesar do algoritmo Cap-DPPM e SSCAM apresentarem valores de BBR sob cargas baixas, estes algoritmos são os que possuem maior número de saltos para o caminho primário. Isto acontece pois a escolha do núcleo e espectro é realizada após a escolha dos enlaces. O algoritmo BARTMAN mantém caminhos primários com baixo número médio de saltos para todas as cargas simuladas, mesmo produzindo menores valores de BBR que os outros algoritmos.

A Figura 4.13 mostra o número médio de saltos do caminho de *backup* gerado para a topologia USA. Como esperado, os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com quase três vezes mais número de saltos do que os caminhos que não o utilizam. No entanto, quando observado o BBR gerado por esses algoritmos, percebe-se que isto não produz tanto impacto na BBR gerada. Isto acontece pois os algoritmos que utilizam *p-cycle* produzem um maior compartilhamento do que os algoritmos que compartilham caminhos. O algoritmo

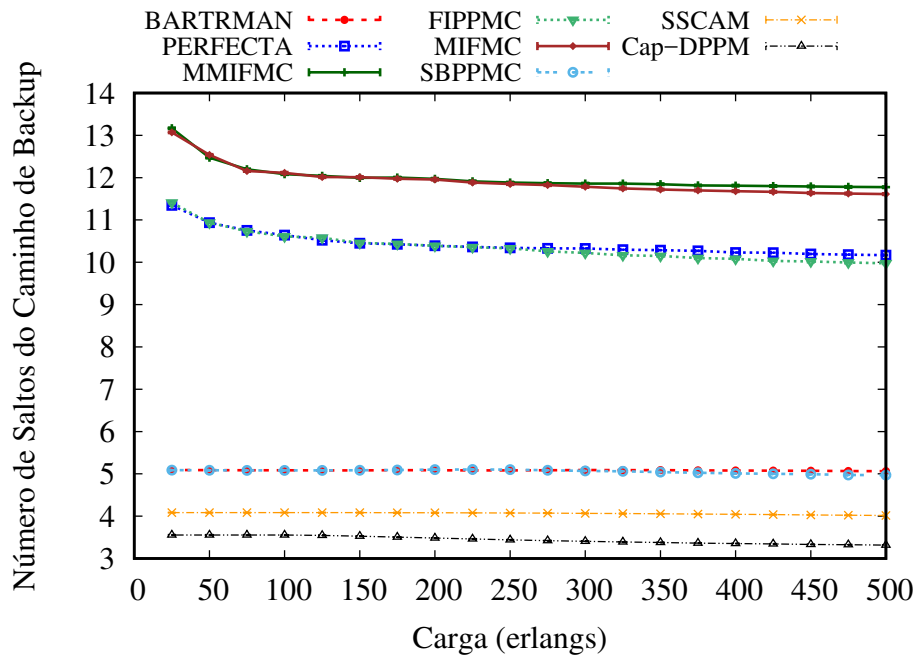


Figura 4.13: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia USA.

BARTRMAN produz número médio de saltos do caminho de *backup* uma unidade maior que o algoritmo SSCAM e mais que uma unidade maior que o algoritmo Cap-DPPM, no entanto isto não se reflete no BBR gerado por estes algoritmos. Os algoritmos SBPPMC, FIPPMC e MIFMC geram caminhos com número de saltos similares aos caminhos gerados pelos algoritmos BARTRMAN, PERFECTA e MMIFMC, respectivamente.

A Figura 4.14 mostra a probabilidade de bloqueio para a topologia NSF. Enquanto o Cap-DPPM inicia o bloqueio de requisições sob cargas de 100 erlangs, SSCAM inicia o bloqueio de requisições apenas sob cargas de 200 erlangs. Os algoritmos FIPPMC e MIFMC iniciam o bloqueio de requisições sob cargas de 125 erlangs e o algoritmo SBPPMC inicia o bloqueio de requisições sob cargas de 150 erlangs. Por sua vez, os algoritmos PERFECTA e MMIFMC iniciam o bloqueio de requisições apenas sob cargas de 300 e 325 erlangs, respectivamente. O algoritmo BARTRMAN como para topologia USA produz o menor bloqueio de requisições. O algoritmo BARTRMAN inicia o bloqueio de requisições apenas sob cargas de 375 erlangs. Sob cargas de 200 erlangs, o algoritmo Cap-DPPM produz valores de BBR três ordens de magnitude maior quando comparado com os valores produzidos pelo algoritmo SSCAM devido ao compartilhamento de caminhos de *backup* utilizado pelo algoritmo SSCAM. Apesar dos algoritmos SBPPMC, FIPPMC e MIFMC utilizarem modulação fixa, estes algoritmos produzem menores valores de BBR que os algoritmos Cap-DPPM que utiliza modulação adaptativa. Isto ocorre pois o algoritmo Cap-DPPM utiliza grande quantidade de recursos para proteção, um vez que os recursos de proteção não são compartilhados. Os algoritmos PERFECTA e MMIFMC produzem valores de BBR semelhantes pois a baixa conectividade desta topologia dificulta a criação de *p-cycles* transzonais. Por outro lado, o algoritmo MMIFMC gera mais *p-cycles* que utilizam os mesmos enlaces do caminho como o algoritmo PERFECTA. Sob cargas de

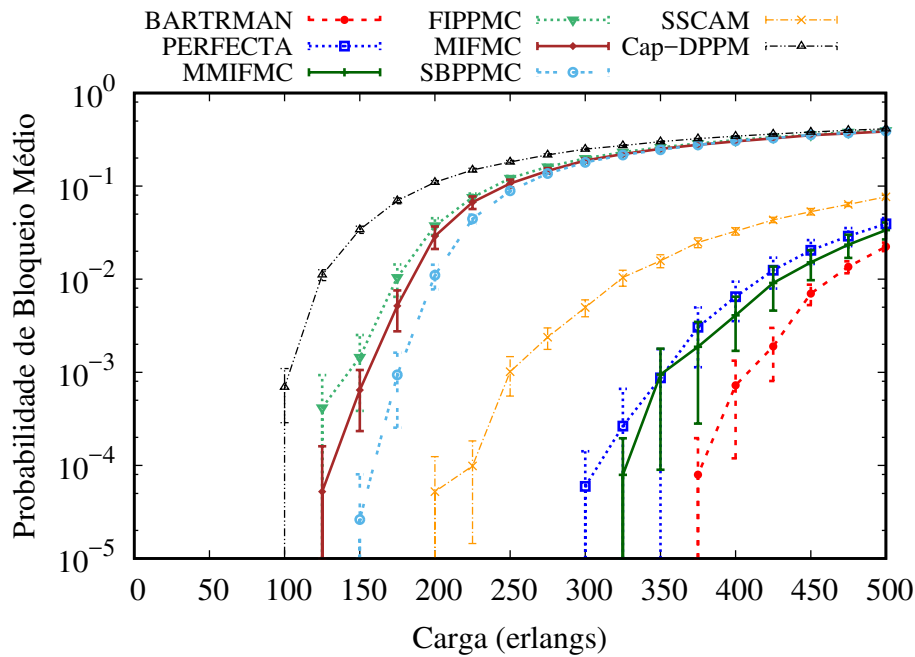


Figura 4.14: Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.

375 erlangs o algoritmo BARTRMAN produz valores de BBR com uma ordem de magnitude menor quando comparado com os valores de BBR produzidos pelos algoritmos PERFECTA e MMIFMC, desde que a baixa conectividade da topologia NSF torna difícil a criação de p-cycles.

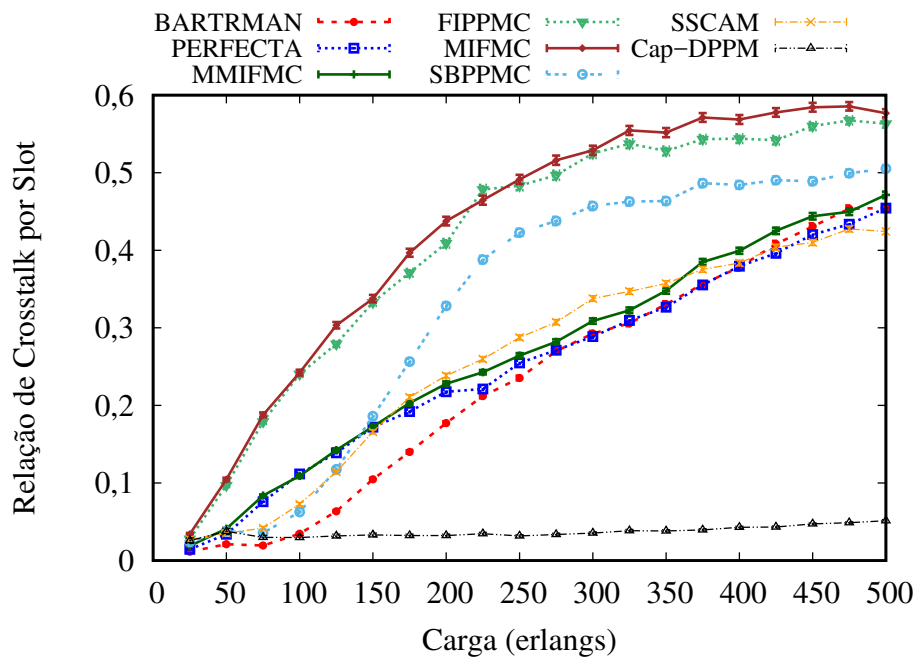


Figura 4.15: *crosstalk* por *slot* dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.

A Figura 4.15 mostra a relação de *crossstalk* por *slot* gerado para a topologia NSF. Os valores de CpS gerados pelo algoritmos Cap-DPPM iniciam em 0,02 e permanecem quase constante com o aumento da carga. Os valores de CpS gerados pelos algoritmos SBPPMC, FIPPMC e MIFMC iniciam em 0,02 e incrementam até 0,51, 0,56 e 0,57, respectivamente, conforme a carga aumenta. Estes algoritmos produzem os maiores valores de CpS pois a utilização de modulação fixa impacta diretamente na utilização de recursos da rede. Os valores de CpS gerados pelos algoritmos PERFECTA e MMIFMC iniciam em 0,02 e incrementam até 0,45 e 0,46, respectivamente, conforme a carga é aumentada. O valores de CpS gerados pelo algoritmo BARTMAN iniciam em 0,01 e incrementam até 0,45 conforme a carga é incrementada. O valores de CpS gerados pelo algoritmo SSCAM iniciam em 0,03 e incrementam até 0,42, conforme a carga é aumentada. Até carga de 100 erlangs o algoritmo BARTMAN produz os menores valores de CpS gerados impactando na menor BBR produzida. Os valores de CpS gerados pelo algoritmo PERFECTA são sempre menores que os valores gerados pelo algoritmo MMIFMC, isto ocorre devido o menor BBR gerado pelo algoritmo MMIFMC.

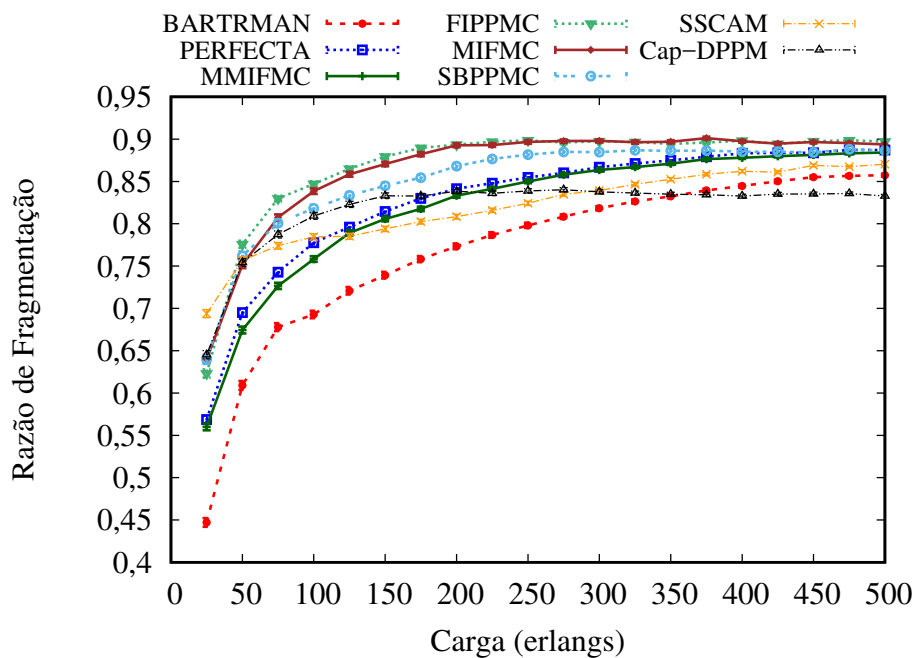


Figura 4.16: Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia NSF.

A Figura 4.16 mostra a razão de fragmentação gerada para a topologia NSF. Até cargas de 325 erlangs, o algoritmo BARTMAN produz a menor razão de fragmentação, isto explica o baixo BBR gerado por este algoritmo. Os algoritmos SBPPMC, FIPPMC e MIFMC que utilizam modulação fixa produzem maior razão de fragmentação do que os algoritmos que utilizam modulação adaptativa. Isto acontece pois a utilização de modulação adaptativa, permite o envio de um maior número de bits por símbolo, e conseqüentemente, diminui a quantidade de slots demandada, permitindo alocação de caminhos em fragmentos de slots menores. A partir de 325 erlangs, o algoritmo BARTMAN produz uma razão de fragmentação mais próxima a dos outros algoritmos simulados, isto explica a aproximação entre os BBRs gerados por esses algoritmos. A razão de fragmentação gerada pelo algoritmo PERFECTA é sempre maior que a

razão de fragmentação gerada pelo algoritmo MMIFMC, isto ocorre pois o maior tamanho dos *p-cycles* gerados pelo algoritmo MMIFMC promove maior compartilhamento dos caminhos de *backup* gerando menos desestabelecimento de conexões. O algoritmo BARTRMAN produz uma menor razão fragmentação que o algoritmo SSCAM para todas as cargas simuladas, isto justifica a menor BBR gerada pelo algoritmo BARTRMAN.

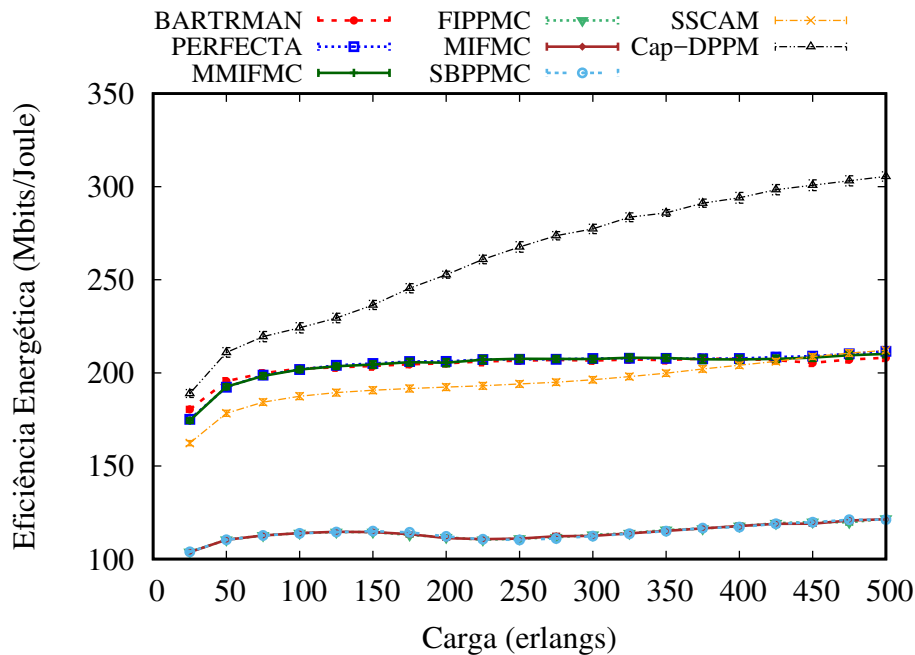


Figura 4.17: Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.

A Figura 4.17 mostra a eficiência energética gerada para a topologia NSF. O algoritmo Cap-DPPM produz a maior eficiência energética em relação aos algoritmos simulados, isto acontece pois o algoritmo Cap-DPPM produz os maiores valores de BBR. Os algoritmos SBPPM, FIPPMC e MIFMC produzem eficiência energética semelhantes. Além disso, estes algoritmos produzem os menores valores de eficiência energética, pois a utilização de modulação fixa, subutiliza o espectro realizando maior consumo energético. Dentre os algoritmos que utilizam modulação adaptativa o algoritmo SSCAM produz a menor eficiência energética em relação aos algoritmos simulados mesmo produzindo valores de BBR maiores que os produzidos pelos algoritmos BARTRMAN, PERFECTA, MMIFMC, isto ocorre pois este algoritmo produz caminhos primários com os maiores número médio de saltos. Para cargas de até 400 erlangs, os algoritmos BARTRMAN, PERFECTA e MMIFMC produzem eficiência energética semelhantes entre si e maiores do que o algoritmo SSCAM, apesar do algoritmo SSCAM produzir maiores valores de BBR. Sob cargas maiores que 400 erlangs, o algoritmo SSCAM produz eficiência energética semelhante aos algoritmos BARTRMAN, PERFECTA e MMIFMC, pois, enquanto o algoritmo SSCAM produz altos valores BBR, os outros algoritmos mantém valores de BBR mais baixo.

A Figura 4.18 mostra o número médio de saltos do caminho primário gerado para a topologia USA. Como nas topologias anteriores, os algoritmos PERFECTA, BARTRMAN e MMIFMC que utilizam o multigrafo possuem número médio de saltos do caminho primário se-

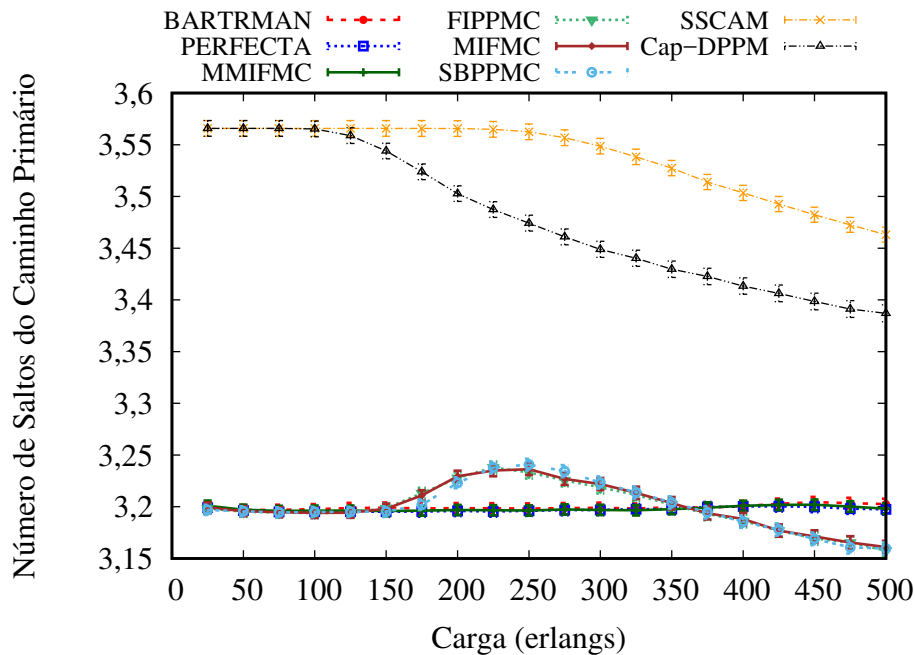


Figura 4.18: Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTRMAN e MMIFMC em função da carga da rede para topologia NSF.

melhantes para qualquer carga simulada. Sob cargas baixas, os algoritmos SBPPMC, FIPPMC e MIFMC produzem caminhos primários semelhantes. Além disso, os algoritmos que utilizam o multigrafo possuem sempre menor número médio de saltos que os caminhos que não utilizam o multigrafo. Sob baixas cargas os algoritmos Cap-DPPM e SSCA possuem número médio de saltos do caminho primário semelhantes. Sob altas cargas o algoritmo SSCAM produz o maior número médio de saltos do caminho primário quando comparado com os outros algoritmos simulados.

A Figura 4.19 mostra o número médio de saltos do caminho de *backup* gerado para a topologia NSF. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com duas vezes mais número médio de saltos que os caminhos que não utilizam. Os algoritmos PERFECTA e MMIFMC que usam o *p-cycle* diferem em aproximadamente uma unidade no número médio de saltos do caminho de *backup* devido ao uso de *p-cycles* transzonais para o algoritmo MMIFMC. Os algoritmos BARTRMAN e SSCAM possuem número médio de saltos do caminho de *backup* apesar do maior BBR produzido pelo algoritmo SSCAM. O algoritmo Cap-DPPM produz o menor número médio de saltos do caminho de *backup* e ainda assim produz um maior BBR, evidenciando as vantagens de utilizar compartilhamento. Quando comparado os algoritmos SBPPMC, FIPPMC e MIFMC que utilizam modulação fixa e os respectivos algoritmos BARTRMAN, PERFECTA e MMIFMC que utilizam modulação adaptativa, conclui-se que a modulação adaptativa não produz muita diferença número médio de saltos dos caminhos de *backup*.

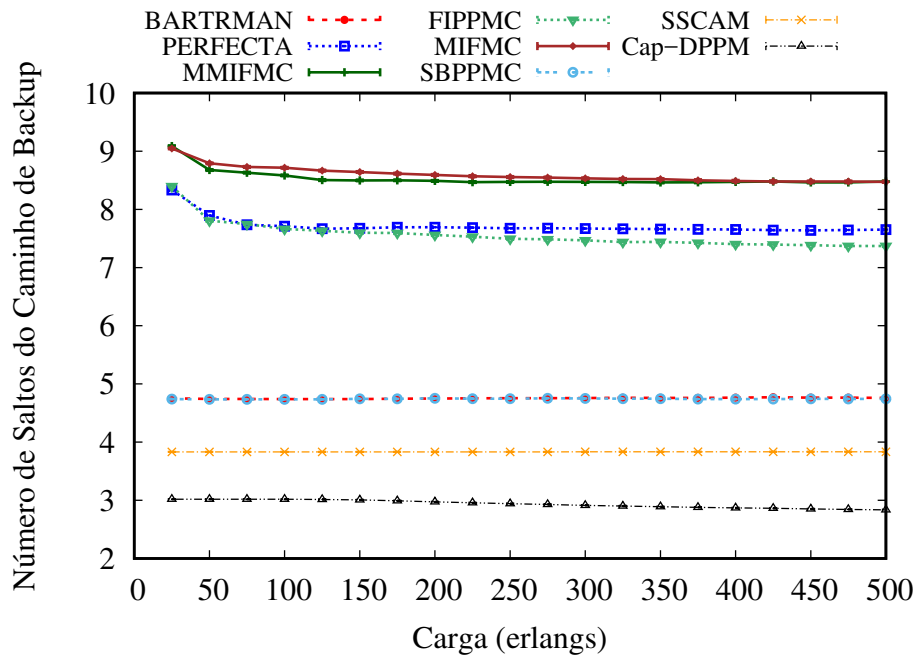


Figura 4.19: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPPM, SSCAM, SBPPMC, FIPPMC, MIFMC, PERFECTA, BARTMAN e MMIFMC em função da carga da rede para topologia NSF.

4.7 Síntese do capítulo

Neste capítulo, apresentaram-se os algoritmos PERFECTA, BARTMAN e MMIFMC para proteção de redes ópticas elásticas com multiplexação espacial. Os três algoritmos utilizam o multigrafo para encontrar caminhos através de topologias com múltiplos núcleos apresentada no capítulo anterior (multigrafo). Além disso, os três algoritmos utilizam modulação adaptativa considerando a distância entre os nós fonte-destino das requisições tanto para o caminho primário quanto para o caminho de backup.

Os algoritmos apresentados são extensões dos algoritmos anteriores utilizando modulação adaptativa. Os resultados mostraram uma grande redução na BBR produzida para os algoritmos propostos neste capítulo com relação às apresentadas no capítulo anterior. Os valores de BBR produzidos pelos algoritmos propostos evidenciam o benefício de escolher conjuntamente a rota, núcleo e modulação quando comparados à escolha em diferentes etapas.

Em geral a relação de fragmentação apresentou grande influência do BBR gerado pelos algoritmos simulados. Os algoritmos que produziram maior BBR tenderam a produzir a maior eficiência energética. O emprego de modulação adaptativa nos algoritmos influenciou em um maior consumo energético.

A utilização de diferentes formatos de modulação melhora a utilização dos recursos disponíveis, de forma que caminhos menores suportam maior transferência de dados.

Capítulo 5

Agregação de tráfego e sobreposição de espectro para proteção de redes ópticas elásticas com multiplexação espacial

As redes ópticas elásticas com multiplexação espacial podem oferecer comunicações ópticas dinâmicas e flexíveis alocando canais ópticos de largura de banda variáveis. Ao utilizar agregação de tráfego óptico nas conexões, é possível reduzir a sobrecarga do espectro causada por bandas de guardas e, assim, melhorar a eficiência espectral das redes. A agregação de tráfego óptico beneficia-se da flexibilidade proporcionada pelos comutadores agregando múltiplos caminhos ópticos em um único transmissor e comutando caminhos ópticos em conjunto. Este agrupamento é realizado de modo transparente, sem conversão do domínio óptico para elétrico, e é denominado túnel óptico.

Em redes ópticas WDM, o compartilhamento de espectro entre canais de comprimento de onda em caminhos de *backup* não é permitido. Usando a possibilidade da elasticidade dos transponders, os algoritmos propostos neste capítulo consideram uma nova oportunidade de compartilhamento de espectro. Nesses algoritmos, se duas conexões possuírem caminhos primários disjuntos, e seus caminhos de *backup* percorrerem dois caminhos ópticos adjacentes em uma fibra, os dois caminhos ópticos de *backup* poderão sobrepor (compartilhar) o espectro.

O *slot* de frequência sobreposto é compartilhado entre caminhos ópticos adjacentes no domínio do tempo: ele é ocupado por no máximo um dos caminhos ópticos adjacentes por vez. A utilização de sobreposição de espectro não obrigatoriamente necessita da agregação de tráfego, uma vez que o transponder elástico pode ajustar a posição do slot de banda de guarda utilizado, podendo, assim, aumentar a largura do espectro utilizado por um caminho de *backup* enquanto diminui outro caminho de *backup* adjacente, satisfazendo a proteção da requisição.

Este capítulo avalia o impacto de utilizar agregação de tráfego e sobreposição de espectro para proteção em redes ópticas elásticas com multiplexação espacial, através de quatro novos algoritmos. As técnicas de agregação de tráfego e sobreposição de espectro utilizadas já foram propostas na literatura, contudo não haviam sido estudada em EONs SDM até então. Os algoritmos propostos diferenciam-se na forma de proteção. Os algoritmos apresentados, neste capítulo, estendem os algoritmos apresentados no Capítulo 3, adicionando aos algoritmos a agregação de tráfego e a sobreposição de espectro nos caminhos de *backup*. Como as requisições são dinâmicas, a agregação de tráfego é realizada através de um fluxo adicional que é

agregado em um caminho óptico existente, aumentando o número de slots utilizados, contudo, sem a necessidade de banda de guarda [54]. Neste cenário, o tempo gasto para criar o túnel óptico não foi estudado nem considerado. Além disso, para facilitar a escolha de caminhos que utilizam agregação de tráfego, o multigrafo considera os *slots* de banda de guarda dos caminhos ópticos que podem ser agregados como livre. Desta forma é possível encontrar o novo caminho, já considerando a agregação de tráfego. Se o novo caminho encontrado utilizar agregação, o túnel óptico será criado com os dois caminhos.

Este capítulo está organizado conforme segue. Na Seção 5.1, apresenta-se uma revisão bibliográfica. Na seção 5.2 apresenta-se a notação utilizada nos algoritmos apresentados nesse capítulo. Nas Seções 5.3, 5.4, 5.5 e 5.6 descrevem-se, respectivamente, o algoritmo *Spectrum, Backuping, Routing and Core assignment* (SBRC), o algoritmo *Spectrum overlap, Traffic grooming and FIPP P-cycle* (STOP), o algoritmo *FIPP p-cycle, Routing, Spectrum and Core Assignment* (FRSCA) e o *Sharing Slot and Straddling p-Cycle FIPP* (SSSP). Na Seção 5.7, compara-se o desempenho dos algoritmos propostos e de outros dois algoritmos. Finalmente, na Seção 5.8, apresenta-se a síntese do capítulo.

5.1 Trabalhos relacionados

Nesta seção, apresentam-se os principais trabalhos que utilizam agregação de tráfego presentes na literatura.

Liu *et al.* [55] propuseram uma nova técnica para a proteção compartilhada em redes ópticas elásticas que proporciona o compartilhamento de *slots* de espectro entre caminhos ópticos de *backup* se os caminhos primários forem disjuntos. A abordagem de proteção proposta, chamada *elastic separate-protection-at-connection* (ESPAC) fornece proteção fim a fim.

Foi proposto em [116] um modelo de grafo auxiliar de três camadas em que ao se ajustar os pesos de ponta das arestas dos gráficos, é possível utilizar várias políticas de tratamento de tráfego para diferentes fins em redes ópticas elásticas. Foram propostos, também, dois esquemas de reserva de espectro para utilizar eficientemente a capacidade de um transpônder. Além disso, diferentes políticas de agregação de tráfego foram comparadas.

Os autores em [64] apresentam um algoritmo para fornecer proteção de caminho, usando *p-cycle* FIPP, agregação de tráfego e sobreposição de espectro em redes ópticas elásticas. O algoritmo proposto foi comparado com outros na literatura.

Costa *et al.* em [19] propõem um algoritmo para resolver o problema de roteamento, nível de modulação e alocação de espectro em EONs através do uso da agregação de tráfego elétrica e óptica associado ao controle da modulação espectral no cenário de tráfego dinâmico. O algoritmo procura realizar a maior quantidade de agregação óptica possível usando os níveis de modulação mais altos através de múltiplos saltos na topologia virtual para redes ópticas elásticas.

Em [92], os autores introduziram uma nova arquitetura de nó em EONs que permite a realização da agregação no nó fonte de forma econômica. Os resultados obtidos demonstram uma melhoria considerável na utilização do espectro da rede, bem como uma redução significativa no número de transmissores necessários por nó.

Os autores em [111] abordam o problema de agregação de tráfego em EONs levando em

consideração a distância com o objetivo de maximizar a eficiência espectral considerando a restrição de alcance da transmissão. No algoritmo, as demandas de tráfego com mesma origem que compartilham o mesmo enlace são agregadas opticamente e comutadas pelo meio óptico sem a utilização de bandas de guarda.

A Tabela 5.1 apresenta uma comparação entre os trabalhos relacionados e os algoritmos propostos neste capítulo. Os trabalhos foram categorizados em relação à proteção, ao uso de multiplexação espacial, ao uso de agregação de tráfego (TG), ao uso de sobreposição de espectro (SE), e ao uso do p -cycle para fornecer proteção. O uso de proteção utiliza caminhos de *backup* compartilhados e, por fim, se provê proteção contra dupla falha.

Tabela 5.1: Comparação dos trabalhos relacionados com os algoritmos SBRC, STOP, FRSCA e SSSPF.

Abordagem	Proteção	SDM	TG	SE	p -Cycle	SBPP	Dupla falha
Liu <i>et al.</i> [55]	Sim	Não	Sim	Sim	Não	Sim	Não
Zhang <i>et al.</i> [116]	Não	Não	Sim	Não	Não	Não	Não
Oliveira <i>et al.</i> [64]	Sim	Não	Sim	Sim	Sim	Não	Não
Costa <i>et al.</i> [19]	Não	Não	Sim	Não	Não	Não	Não
Khodashenas <i>et al.</i> [92]	Não	Não	Sim	Não	Não	Não	Não
Ye <i>et al.</i> [111]	Não	Não	Sim	Não	Não	Não	Não
SBRC	Sim	Sim	Sim	Sim	Não	Sim	Não
STOP	Sim	Sim	Sim	Sim	Sim	Não	Não
FRSCA	Sim	Sim	Sim	Sim	Sim	Não	Não
SSSPF	Sim	Sim	Sim	Sim	Sim	Não	Sim

5.2 Notação

A notação matemática a seguir será usada neste capítulo. A notação adiciona à notação do Capítulo 3. As principais diferenças são referentes a modelagem de agregação de tráfego e de sobreposição de espectro:

$\Delta(G, C, b) = \{\widetilde{G}'_{n,b}\}$: função que produz todos os σ grafos de G considerando as possíveis agregações de tráfego;

$\widetilde{G}'_{n,b} = (\widetilde{V}, \widetilde{E}, \widetilde{W})$: o n -ésimo grafo tal que \widetilde{E} é o conjunto de arestas conectando $\{\widetilde{u}, \widetilde{v}\} \in \widetilde{V}$ e \widetilde{W} é conjunto de custos associados a \widetilde{E} . As arestas em \widetilde{E} correspondem ao mapeamento de b arestas em G iniciando na n -ésima aresta;

$\widetilde{G}''_{n,b} = (\widetilde{V}, \widetilde{E}, \widetilde{W})$: o n -ésimo grafo tal que \widetilde{E} é o conjunto de arestas conectando $\{\widetilde{u}, \widetilde{v}\} \in \widetilde{V}$ e \widetilde{W} é conjunto de custos associados a \widetilde{E} . As arestas em \widetilde{E} correspondem ao mapeamento de b arestas em G iniciando na n -ésima aresta;

$\kappa(G, C, b, P_n) = \{\widetilde{G}''_{n,b}\}$: função que produz todos os grafos de G , considerando que os *slots* de proteção podem ser sobrepostos, desde que os caminhos de trabalho (P_n) que são protegidos pelos slots dos caminhos de *backup* sejam fisicamente disjuntas (compartilhamento de espectro);

$\rho(P_n, Z, r(s, d, b))$: função que procura o menor caminho de *backup* em Z_n em que P_{Z_n} são enlaces disjuntos de P_n e satisfazem a requisição de banda b e possuem nós finais s e d ;

Z_n : sequência de $\widetilde{G}''_{n,b}$ tal que o nó s é o menor nó ordenado e d é o maior nó ordenado;

- P_{Z_n} : conjunto de todos os caminhos protegidos pelo caminho de *backup* Z_n ;
 H_{P_n} : conjunto de todos os *slots* usado pelo caminho P_n ;
 $H_{P_{Z_n}}$: conjunto de todos os *slots* usados por todos os caminhos protegidos pelo caminho de *backup* Z_n ;
 $Z = \{Z_n\}$: conjunto de todos os caminhos de estabelecidos;
 $\zeta(\widetilde{G}''_{n,b,m}, P_n, r(s, d, b))$: função que produz menor caminho entre s e d em $\widetilde{G}''_{n,b}$, considerando que $H_{P_{Z_n}}$ é disjunto de H_{P_n} ;
 $W(Z_n)$: o peso do caminho de *backup* Z_n (a soma dos pesos de todas as arestas na sequencia);
 $W_{Z_{s,d}}$ = peso do caminho de *backup* que protege o caminho entre s e d ;
 T_n : p -cycle contendo os vértices u e v e arestas correspondendo ao mapeamento de b arestas do multigrafo G ;
 T : conjunto de todos os p -cycles em G ;
 P_{T_n} : conjunto de todos os caminho protegidos pelo p -cycle T_n ;
 H_{P_n} : conjunto de todos os *slots* usados pelo caminho P_n ;
 $H_{P_{T_n}}$: conjunto de todos os *slots* usados por todos os caminhos protegidos pelo T_n ;
 $T = \{T_n\}$: conjunto de todos os p -cycles estabelecidos;
 $\varpi(P_n, T, r(s, d, b))$: função que procura um p -cycle em T que P_{T_n} são caminhos disjuntos de P_n e satisfazem a requisição b e que s e d fazem parte do p -cycle;
 $\gamma(\widetilde{G}''_{n,b,m}, P_n, r(s, d, b))$: função que produz o menor p -cycle entre s e d em $\widetilde{G}''_{n,b}$, considerando que $H_{P_{T_n}}$ é disjunto de H_{P_n} ;
 $W(T_n)$: o peso do p -cycle T_n (a soma dos pesos de todos as arestas na sequencia);
 $W_{T_{s,d}}$ = peso do p -cycle que protege o caminho primário entre os nós s e d ;
 $\varrho(P_n, T, r(s, d, b))$: função que procura um p -cycle transzonal a P_n em T , ou seja, P_{T_n} e T_n são caminhos disjuntos de P_n e satisfazem a requisição b e que s e d fazem parte do p -cycle;
 $\varphi(\widetilde{G}''_{n,b}, P_n, r(s, d, b))$: função que produz o menor p -cycle transzonal ao caminho P_n em que $H_{P_{T_n}}$ são *slots* disjuntos de H_{P_n} and satisfazem a requisição de banda b e que o p -cycle produzido T_n é disjunto do caminho P_n ;

5.3 O algoritmo SBRC

No algoritmo *Spectrum, Backuping, Routing and Core assignment* (SBRC), para cada conexão é atribuído um caminho primário e um caminho de *backup* . Neste algoritmo o caminho primário utiliza sempre que possível agregação de tráfego e o caminho de *backup* utiliza sempre que possível agregação de tráfego e sobreposição de espectro. O algoritmo SBRC estende o algoritmo SBPPMC, diferenciando-se na criação dos caminhos, agregando tráfego e sobrepondo espectro entre caminhos de *backup* sempre que possível.

No algoritmo SBRC (Algoritmo 8), a linha 1 transforma o multigrafo em $C \times (N - b + 1)$ grafos considerando a possibilidade de realizar agregação de tráfego entre caminhos de mesma origem, deste modo facilitando a busca de caminhos nos grafos gerados. A busca do caminho primário (linha 2) é realizada utilizando o algoritmo de Dijkstra para os grafos gerados, sendo escolhido o caminho com menor custo (linha 3). Se não for possível encontrar um caminho que respeite a restrições de contiguidade e continuidade para a demanda b (linha 5), a conexão é

Algoritmo 8 SBRC**Entrada:** $G, r(s, d, b)$ **Saída:** Caminho primário e Caminho de *backup*

```

1:  $\Delta(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\widetilde{G}'_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} = \infty$  então
6:   bloqueia requisição  $(r(s, d, b))$ 
7: Senão
8:   Se  $\exists \rho(P_n, Z, r(s, d, b))$  então
9:     estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
10:  Senão
11:     $\kappa(G, C, b, P_n)$ 
12:     $(W(Z_n), Z_n) = \zeta(\widetilde{G}''_{n,b}, P_n, r(s, d, b))$ 
13:     $W_{Z_{s,d}} = W(Z_n) \mid \forall i W(Z_n) \leq W(Z_i)$ 
14:    Se  $W_{Z_{s,d}} = \infty$  então
15:      bloqueia requisição  $(r(s, d, b))$ 
16:    Senão
17:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
18:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in Z_i$ 
19:    Fim Se
20:  Fim Se
21: Fim Se

```

bloqueada (linha 6). Caso contrário, um caminho de *backup* será buscado dentre os caminhos de *backup* já estabelecidos para outras requisições. Se existir um caminho de *backup* disponível (linha 9), a requisição é estabelecida (linha 10). Se não existir nenhum caminho de *backup* que possa proteger a requisição, então um novo caminho precisa ser criado. Na linha 11, o multigrafo é transformado em $C \times (N - b)$ grafos, considerando a agregação de tráfego e a sobreposição de *slots*. A sobreposição de espectro possibilita que *slots* que estão protegendo outras conexões sejam utilizados em novas conexões, desde que, respeitando as regras de compartilhamento. A linha 12 utiliza o algoritmo de Dijkstra e os grafos gerados na linha 11 para buscar um menor caminho de *backup* nos grafos. A linha 13 escolhe o menor caminho dos grafos para caminho de *backup*. Se o peso do menor caminho for ∞ , significa que um caminho de *backup* não pode ser encontrado e a requisição é bloqueada (linha 15). Caso contrário, na linha 17, o caminho primário bem como o caminho de *backup* são estabelecidos.

A complexidade do algoritmo SBRC foi analisada. A complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para o caminho de *backup*, a complexidade de transformar o multigrafo em grafos é $O(E + V)$ para geração de novos grafos, considerando a agregação e a sobreposição de espectro; em seguida o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos. Logo, no pior caso, o algoritmo SBRC tem complexidade de $2 \times (E + V + C \times (N - b) \times (E + V \log V))$. Como N, C e b podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

5.4 O algoritmo STOP

Nesta seção, propõe-se um algoritmo chamado *Spectrum overlap, Traffic grooming and FIPP P-cycle* (STOP). O algoritmo provê proteção para redes ópticas elásticas utilizando multiplexação espacial através de *p-cycles* FIPP, agregação de tráfego e sobreposição de espectro. Neste algoritmo, os *p-cycles* são criados compartilhando espectro com outros *p-cycle* desde que os caminhos primários protegidos por eles utilizem fibras disjuntas. Além disso, a agregação de tráfego também é utilizada. O algoritmo STOP é uma extensão do algoritmo FIPPMC. A diferença entre o algoritmo FIPPMC e o algoritmo STOP é a criação do caminho primário e do *p-cycle*.

Algoritmo 9 STOP

Entrada: $G, r(s, d, b)$

Saída: Caminho primário e Caminho de *backup*

- 1: $\Delta(G, C, b)$
 - 2: $(W(P_n), P_n) = \delta(\widetilde{G}'_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$
 - 3: $W_{P_{s,d}} = W(P_n) | \forall i W(P_n) \leq W(P_i)$
 - 4: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$
 - 5: **Se** $W_{P_{s,d}} = \infty$ **então**
 - 6: bloqueia requisição $(r(s, d, b))$
 - 7: **Senão**
 - 8: **Se** $\exists \varpi(P_n, T, r(s, d, b))$ **então**
 - 9: estabelece requisição $(r(s, d, b))$ **como** P_n **e** T_n
 - 10: **Senão**
 - 11: $\kappa(G, C, b, P_n)$
 - 12: $(W(T_n), T_n) = \gamma(\widetilde{G}''_{n,b}, P_n, r(s, d, b))$
 - 13: $W_{T_{s,d}} = W(T_n) | \forall i W(T_n) \leq W(T_i)$
 - 14: **Se** $W_{T_{s,d}} = \infty$ **então**
 - 15: bloqueia requisição $(r(s, d, b))$
 - 16: **Senão**
 - 17: estabelece requisição $(r(s, d, b))$ **como** P_n **e** T_n
 - 18: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$
 - 19: **Fim Se**
 - 20: **Fim Se**
 - 21: **Fim Se**
-

No algoritmo STOP, como nos algoritmos anteriores, a linha 1 transforma o multigrafo em $C \times (N - b + 1)$ grafos considerando a agregação de tráfego, a linha 2 computa o menor caminho para os $\widetilde{G}'_{n,b}$ grafos utilizando o algoritmo de Dijkstra e atribui o peso a ele (linha 3). Se o peso do menor caminho for ∞ ; isto significa que nenhum caminho que satisfaz a requisição pode ser encontrado, então a linha 6 bloqueia a requisição. Caso contrário, um *p-cycle* FIPP para proteger o caminho primário encontrado deve ser buscado entre os *p-cycles* estabelecidos ativos (linha 8), pois um *p-cycle* FIPP só é válido se os nós finais do caminho estiverem no *p-cycle*. Além disso, os *slots* que fazem parte do caminho primário H_{P_n} precisam ser disjuntos de todos os *slots* usados por todos os caminhos protegidos pelo T_n ($H_{P_{T_n}}$). Se um *p-cycle* FIPP que satisfaz a requisição for encontrado, a requisição é estabelecida (linha 9) e as arestas correspondentes ao caminho no multigrafo é configurada para ∞ (linha 10). Se os *p-cycles* ativos não puderem proteger o novo caminho, é necessário criar um novo *p-cycle* FIPP. A linha

11 transforma o multigrafo em $C \times (N - b + 1)$ grafos, considerando a sobreposição de *slots* de espectro para proteger caminhos primários. Na sobreposição de *slots* de espectro, *slots* no grafo são configurados como disponíveis se estão disponíveis no multigrafo ou se o *slot* está sendo usado para proteger um caminho disjunto do caminho primário P_n . Na linha 12, o menor *p-cycle* FIPP para proteger o caminho óptico é buscado utilizando os grafos gerados e o algoritmo de Suurballe. Caso o *p-cycle* não possa ser criado (linha 14), a requisição é bloqueada (linha 15). Caso contrário, o caminho primário, bem como, *p-cycle* FIPP (Linha 17) são estabelecidos para satisfazer a solicitação, e as arestas correspondentes têm seu peso alterado para ∞ .

A complexidade do algoritmo STOP é analisada a seguir. A complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma para o *p-cycle* o algoritmo de Suurballe é executado em $C \times (N - b)$ grafos, em que a complexidade de Suurballe é $O(E + V \log V)$. Logo no pior caso, o algoritmo STOP tem complexidade de $E + V + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V)$. Como N , C e b podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

5.5 O algoritmo FRSCA

Nesta seção, propõe-se um algoritmo chamado *FIPP p-cycle, Routing, Spectrum and Core Assignment* (FRSCA). O algoritmo proposto estende o algoritmo 4 (MIFMC), adicionando agregação de tráfego para criação dos caminhos primários e agregação de tráfego e sobreposição de espectro para a criação dos *p-cycles*. No algoritmo FRSCA, todos os caminhos primários são protegidos por um *p-cycle* compartilhado. Neste algoritmo, os *p-cycles* são criados priorizando a criação de *p-cycle* transzonais ao caminho, caso não seja possível, um *p-cycle* não transzonal ao caminho é criado.

O algoritmo FRSCA é introduzido no algoritmo 10. Como nos algoritmos anteriores a linha 1 transforma o multigrafo em $C \times (N - b + 1)$ grafos de modo a facilitar a forma de encontrar o menor caminho entre dois nós. Além disso, considera a agregação de tráfego na criação dos grafos. Após a obtenção dos grafos, a linha 2 utiliza o algoritmo de menor caminho de Dijkstra para encontrar o caminho primário. A linha 3 define o menor caminho e seu peso. Se o peso do menor caminho for ∞ , significa que não foi possível encontrar um caminho e a requisição é bloqueada. Caso contrário, busca-se um *p-cycle* FIPP para cada requisição estabelecida. A linha 8 procura, no conjunto de *p-cycles* FIPP estabelecidos, se existe um *p-cycle* que possa prover proteção para o caminho primário criado. Se houver um *p-cycle*, a linha 8 aceita a requisição e os pesos das arestas do multigrafo correspondente ao caminho são configuradas para ∞ . Como este algoritmo utiliza sobreposição de espectro nos *p-cycles* criados, o *p-cycle* FIPP só é valido para proteger o caminho se os *slots* que fazem parte do caminho primário H_{P_n} forem disjuntos de todos os *slots* usados por todos os caminhos protegidos pelo T_n ($H_{P_{T_n}}$). Se nenhum *p-cycle* FIPP já estabelecido puder proteger o novo caminho, então um novo caminho precisa ser criado. Na criação do *p-cycle*, este algoritmo tenta encontrar um *p-cycle* FIPP transzonal ao caminho, mas, se isso não for possível, o algoritmo tenta encontrar um *p-cycle* FIPP que não seja transzonal ao caminho. Para a criação do *p-cycle* FIPP transzonal ao caminho, a linha 11

Algoritmo 10 FRSCA**Entrada:** $G, r(s, d, b)$ **Saída:** Caminho primário e Caminho de *backup*

```

1:  $\Delta(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\widetilde{G}'_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} = \infty$  então
6:   bloqueia requisição  $(r(s, d, b))$ 
7: Senão
8:   Se  $\exists \varpi(P_n, T, r(s, d, b))$  então
9:     estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
10:  Senão
11:     $\kappa(G, C, b)$ 
12:     $(W(T_n), T_n) = \varphi(\widetilde{G}''_{n,b}, P_n, r(s, d, b))$ 
13:    Se  $W_{T_{s,d}} = \infty$  então
14:       $(W(T_n), T_n) = \gamma(\widetilde{G}''_{n,b}, P_n, r(s, d, b))$ 
15:      Se  $W_{T_{s,d}} = \infty$  então
16:        bloqueia requisição  $(r(s, d, b))$ 
17:      Fim Se
18:    Fim Se
19:    Se  $W_{T_{s,d}} \neq \infty$  então
20:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
21:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
22:    Fim Se
23:  Fim Se
24: Fim Se

```

transforma o multigrafo em $C \times (N - b + 1)$ grafos, considerando a sobreposição de espectro para proteger *slots*, ou seja, *slots* no grafo são configurados como disponíveis se estão disponíveis no multigrafo ou se o *slot* está sendo usado para proteger um caminho disjunto do caminho primário P_n . Se o *p-cycle* FIPP transzonal não puder ser criado, o algoritmo tenta criar um *p-cycle* que utiliza o mesmo enlace do caminho primário (linha 14). Caso nenhum *p-cycle* FIPP possa ser criado, a requisição é bloqueada (linhas 19 a 21).

A complexidade do algoritmo FRSCA foi analisada. A complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma para o caminho de *backup* no pior caso o algoritmo de Suurballe é executado em $C \times (N - b)$ grafos para o *p-cycle* transzonal e $C \times (N - b)$ grafos para o *p-cycle* de menor caminho, em que a complexidade de Suurballe é $O(E + V \log V)$. Logo no pior caso, o algoritmo MIFMC tem complexidade de $E + V + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V)$. Como N , C e b podem ser expressados como constantes a complexidade do algoritmo é $O(E + V \log V)$.

5.6 O algoritmo SSSPF

O algoritmo introduzido nesta seção, denominado *Sharing Slot and Straddling p-Cycle FIPP* (SSSP), decide sobre o estabelecimento de caminhos ópticos em redes protegidas. Diferentemente dos outros algoritmos apresentados nesta Tese no algoritmo SSSPF um caminho óptico é estabelecido, se e somente se, puder ser protegido por um *p-cycle* contra duas falhas simultâneas de enlace. Para isto, é necessário que o caminho de *backup* possa proteger o caminho primário mesmo que quaisquer dois enlaces da rede falhem. Com esse tipo de proteção é exigido muito recurso da rede, e que pelo menos todos os nós tenham grau três, para que haja uma BBR mínima.

Algoritmo 11 SSSPF

Entrada: $G, r(s, d, b)$

Saída: Caminho primário e Caminho de *backup*

- 1: $\Delta(G, C, b)$
 - 2: $(W(P_n), P_n) = \delta(\widetilde{G}'_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$
 - 3: $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$
 - 4: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$
 - 5: **Se** $W_{P_{s,d}} = \infty$ **então**
 - 6: bloqueia requisição ($r(s, d, b)$)
 - 7: **Senão**
 - 8: **Se** $\exists \varrho(P_n, T_{s,d}, r(s, d, b))$ **então**
 - 9: estabelece requisição ($r(s, d, b)$) **como** P_n **e** T_n
 - 10: **Senão**
 - 11: $\kappa(G, C, b)$
 - 12: $(W(T_n), T_n) = \varphi(\widetilde{G}''_{n,b}, P_n, r(s, d, b))$
 - 13: **Se** $W_{T_{s,d}} = \infty$ **então**
 - 14: bloqueia requisição ($r(s, d, b)$)
 - 15: **Senão**
 - 16: estabelece requisição ($r(s, d, b)$) **como** P_n **e** T_n
 - 17: $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$
 - 18: **Fim Se**
 - 19: **Fim Se**
 - 20: **Fim Se**
-

O algoritmo SSSPF é descrito no algoritmo 11. Como nos algoritmos anteriores a linha 1 transforma o multigrafo em $C \times (N - b + 1)$ grafos de modo que cada grafo criado considera a possível agregação de tráfego, baseado nos caminhos a serem estabelecidos, dessa forma facilitando encontrar o menor caminho entre dois nós. Após a obtenção dos grafos a linha 2 utiliza o algoritmo de menor caminho de Dijkstra para encontrar o caminho primário. A linha 3 define o menor caminho e seu peso. Se o peso do menor caminho for ∞ , significa que não foi possível encontrar um caminho e a requisição é bloqueada. Caso contrário, uma técnica de proteção que proteja contra duas falhas simultâneas precisa ser utilizada. Neste algoritmo, um *p-cycle* FIPP transzonal ao caminho primário é selecionado para cada requisição estabelecida. Um *p-cycle* FIPP transzonal ao caminho forma junto ao caminho primário 3 caminhos disjuntos entre os nós fonte e destino, com isso garantindo proteção contra 2 falhas simultâneas. Como no algoritmo anterior, o *p-cycle* FIPP só é válido para proteger o caminho se os *slots* que fazem parte do caminho primário H_{P_n} forem disjuntos de todos os *slots* usados

por todos os caminhos protegidos pelo $T_n (H_{P_{T_n}})$. A linha 7 procura no conjunto de p -cycles FIPP estabelecidos se existe um p -cycle que seja transzonal ao novo caminho P_n . Se houver um p -cycle, a linha 8 aceita a requisição e os pesos das arestas do multigrafo correspondente ao caminho são configuradas para ∞ . Se nenhum p -cycle FIPP já estabelecido possa proteger o novo caminho, então um novo caminho precisa ser criado. Para a criação do p -cycle transzonal ao caminho, a linha 11 transforma o multigrafo em $C \times (N - b + 1)$ grafos, considerando a sobreposição de espectro para proteger $slots$, ou seja, $slots$ no grafo são configurados como disponíveis se estão disponíveis no multigrafo ou se o $slot$ está sendo usado para proteger um caminho disjunto do caminho primário P_n . Na linha 12, o menor p -cycle FIPP para proteger o caminho óptico é buscado utilizando os grafos gerados e o algoritmo de Yen, para manter os p -cycles transzonais ao caminho, p -cycles que utilizam enlaces do caminho são desconsiderados. Caso possa criar um p -cycle FIPP, o p -cycle, bem como o caminho, será estabelecido, e as arestas correspondente no multigrafo G têm os pesos configurados para ∞ (linhas 16 à 17). Caso contrário, a requisição é bloqueada (linha 14).

A complexidade do algoritmo SSSPF é analisada conforme segue: a complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para o p -cycle o algoritmo de Suurballe é executado em $C \times (N - b)$ grafos, em que a complexidade de Suurballe é $O(E + V \log V)$. Logo, no pior caso, o algoritmo SSSPF tem complexidade de $E + V + C \times (N - b) \times (E + V \log V) + C \times (N - b) \times (E + V \log V)$. Como N , C e b podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

5.7 Resultados numéricos

Como nos capítulos anteriores, para avaliar o desempenho dos algoritmos propostos neste capítulo, experimentos de simulação foram realizados utilizando o simulador FlexGridSim [59]. Em cada simulação, 100.000 requisições foram geradas. Utilizou-se o método de replicação independente e adotou-se nível de confiança de 95% para os intervalos de confiança. Diferentes topologias com diferentes conectividades, foram empregadas nas simulações: a Pan-European (Figura 3.2(a)) com 28 nós e 39 enlaces bidirecionais, a USA (Figura 3.2(b)) com 24 nós e 43 enlaces bidirecionais, a NSF (Figura 3.2(c)), com 14 nós e 18 enlaces bidirecionais. As figuras mostram as distancias entre os nós. O espectro foi dividido em 320 $slots$ de espectro e as fibras simuladas possuem 7 núcleos. A carga foi variada entre 25 e 500 erlangs em intervalos de 25 erlangs. Sete tipos de requisições foram utilizadas 25 Gbps, 50 Gbps, 125 Gbps, 200 Gbps, 500 Gbps, 750 Gbps e 1 Tbps. As métricas usadas para avaliar os algoritmos apresentados neste capítulo são as mesmas utilizadas nos dois capítulos anteriores.

Nas figuras, as curvas rotuladas como SBRC mostram os resultados para as redes que utilizam o algoritmo SBRC (Algoritmo 8), as curvas rotuladas como STOP mostram os resultados para as redes que utilizam o algoritmo STOP (Algoritmo 9), as curvas rotuladas como FRSCA mostram os resultados para as redes que utilizam o algoritmo FRSCA (Algoritmo 10) e as curvas rotuladas como SSSPF mostram os resultados para as redes que utilizam o algoritmo SSSPF (Algoritmo 11). Como no capítulo 3, as curvas rotuladas como SSCA encontram o caminho

primário através do algoritmo proposto em [102], utilizando $k = 3$ e o caminho de *backup*, adicionando a capacidade de compartilhamento ao mesmo algoritmo, enquanto as curvas rotuladas como Cap-DPP mostram os resultados para as redes que utilizam o algoritmo proposto em [100], as curvas rotuladas como FIPPMC mostram os resultados para as redes que utilizam o algoritmo FIPPMC (Algoritmo 3), as curvas rotuladas como MIFMC mostram os resultados para as redes que utilizam o algoritmo MIFMC (Algoritmo 4), e as curvas rotuladas como SBPPMC mostram os resultados para as redes que utilizam o algoritmo SBPPMC (Algoritmo 2).

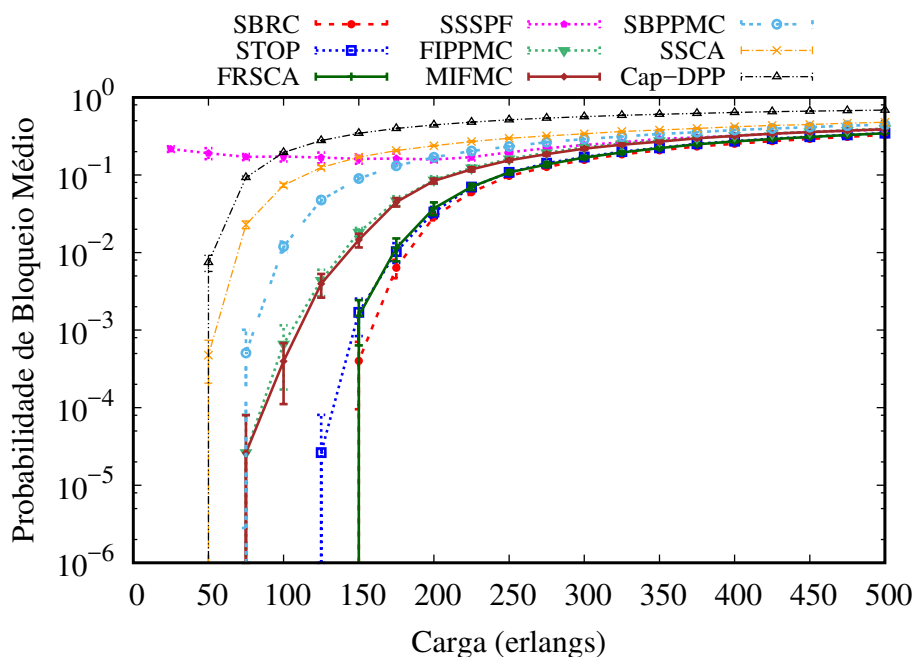


Figura 5.1: Bloqueio de banda dos algoritmos Cap-DPP, SSCA, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.

A Figura 5.1 mostra a probabilidade de bloqueio para a topologia Pan-European. Enquanto os algoritmos SSSPF iniciam o bloqueio de requisições sob cargas de 25 erlangs, os algoritmos Cap-DPP e o SSCA iniciam o bloqueio de requisições sob cargas de 50 erlangs. Os algoritmos SBPPMC, FIPPMC e MIFMC iniciam o bloqueio de requisições sob cargas de 75 erlangs. Por sua vez, os algoritmos SBRC e FRSCA iniciam o bloqueio de requisições sob cargas de 150 erlangs, respectivamente. O algoritmo STOP inicia o bloqueio sob cargas de 125 erlangs. Os algoritmos SBPPMC, FIPPMC e MIFMC produzem maiores valores de BBR que os algoritmos SBRC, STOP e FRSCA. Isto ocorre pois os algoritmos SBPPMC, FIPPMC e MIFMC não utilizam agregação de tráfego e sobreposição de espectro, o que faz com que uma maior quantidade de recursos seja necessária para prover proteção. Os algoritmos Cap-DPP e SSCA produzem os maiores valores de BBR, pois além de não utilizarem agregação de tráfego e sobreposição de espectro, eles utilizam uma forma diferente de encontrar os caminhos utilizados. O algoritmo FRSCA produz BBR apenas sob cargas maiores que 150 erlangs, isso evidencia o benefício de se considerar os critérios de interferência mínima para criar os *p-cycles* durante a escolha da rota de *backup* para topologia com alta conectividade. Os algoritmos FRSCA e STOP produzem comportamento BBR semelhante, devido à alta conectividade da topologia Pan-European. Sob cargas maiores que 150 erlangs, os algoritmos que utilizam o multigrafo na criação dos

caminhos produzem BBR semelhantes, fazendo um bom uso dos recursos da rede. Para cargas maiores que 100 erlangs, o algoritmo Cap-DPP produziu o BBR mais alto, pois ele não usa caminhos compartilhados para executar proteção bem como agregação de tráfego e sobreposição de espectro. Mesmo com uma maior conectividade como a da topologia Pan-European, o algoritmo SSSPF produz um BBR alto, isto ocorre pois, para produzir baixo BBR para um algoritmo que realiza proteção contra dupla falha, é necessária uma topologia em que todos os nós da rede tenham pelo menos grau três. No entanto, sob cargas maiores, o algoritmo SSSPF produziu menor BBR que os algoritmos Cap-DPP e SSCA.

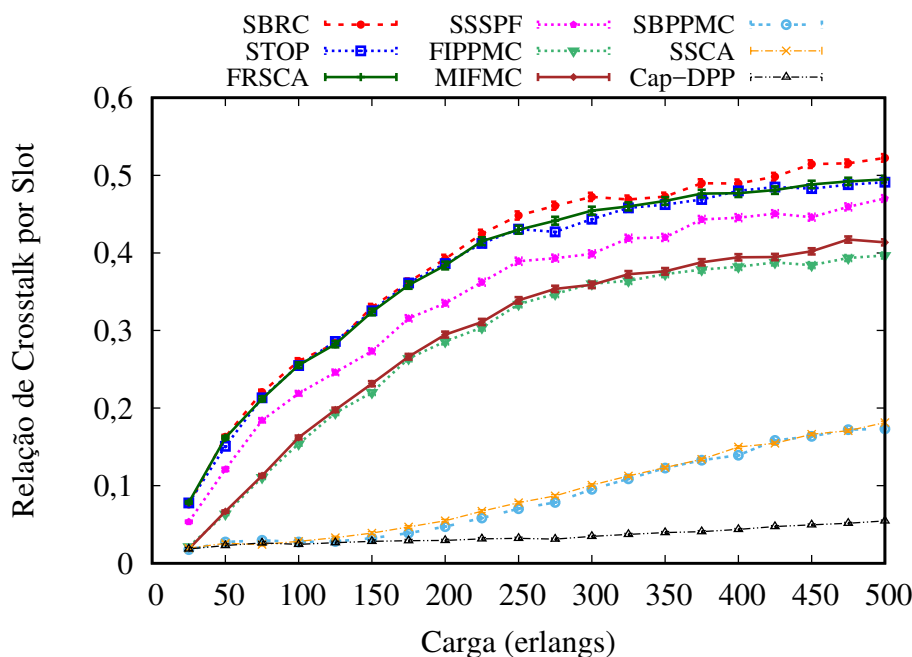


Figura 5.2: *Crosstalk* por *slot* dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.

A Figura 5.2 mostra a relação de *crosstalk* por *slot* gerado para a topologia Pan-European. Os valores de CpS gerados pelos algoritmos FIPPMC e MIFMC iniciam com um valor de 0,02 e aumentam até 0,40. Por sua vez, o algoritmo SBPPMC iniciam com um valor de 0,02 e aumentam até 0,17. Os valores de CpS gerados pelos algoritmos STOP, SBRC e FRSCA iniciam com um valor de 0,07 e aumentam até 0,51. O valores de CpS gerados pelo algoritmo SSSPF inicia com um valor de 0,05 e aumenta até 0,47. O valores de CpS gerados pelo algoritmo SSCA inicia com um valor de 0,02 e aumenta até 0,18. O algoritmo Cap-DPP produz valores de CpS inicia em 0,02 e se mantém quase constante sob cargas mais altas, como consequência do alto bloqueio gerado. Os maiores valores de CpS gerados pelos algoritmos STOP, SBRC e FRSCA quando comparados com os algoritmos SBPPMC, FIPPMC e MIFMC ocorrem devido ao maior número de conexões aceitas como consequência do uso de agregação de tráfego e sobreposição de espectro. O maior bloqueio de conexões do algoritmo SSSPF produz menor CpS com relação aos outros algoritmos que utilizam o multigrafo. Sob baixas cargas apesar do algoritmo SBRC produzir menor valores de BBR do que produz os algoritmos STOP e FRSCA, eles produzem valores de CpS similares. Os algoritmos que utilizam multigrafo possuem maior relação de CpS, como consequência da alta utilização produzida por eles.

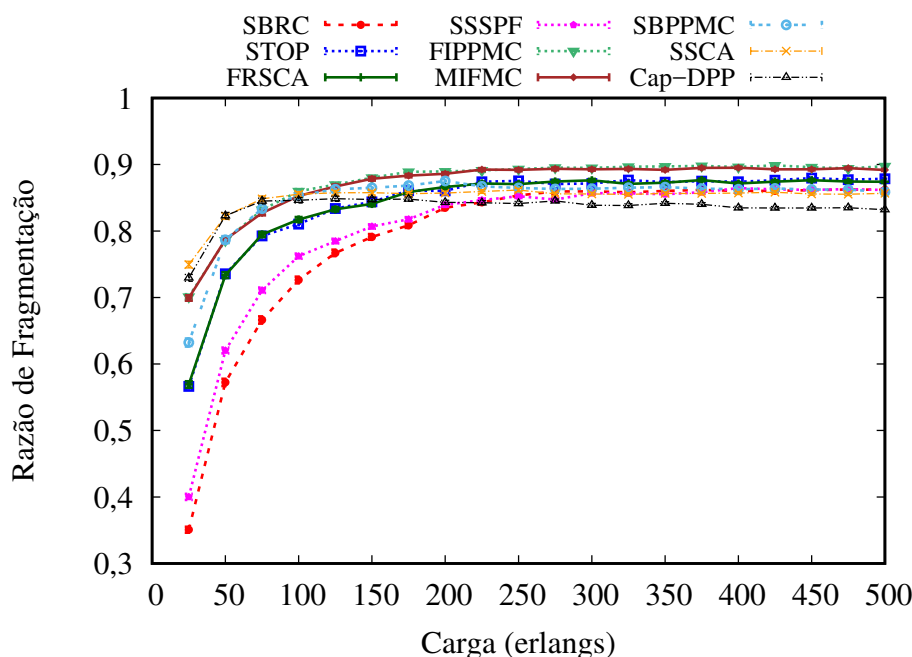


Figura 5.3: Razão de fragmentação dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.

A Figura 5.3 mostra a razão de fragmentação gerada para a topologia Pan-European. Para cargas de até 200 erlangs, o algoritmo SBRC possui a menor razão de fragmentação. Isto acontece pois este algoritmo utiliza menos recursos, devido a utilização do multigrafo e dos caminhos compartilhados. Para cargas de até 75 erlangs, os algoritmos Cap-DPP e SSCA possuem a maior razão de fragmentação justificando o maior bloqueio de requisições que esses algoritmos produzem. Os algoritmos SBPPMC, FIPPMC e MIFMC produzem maior razão de fragmentação do que os algoritmos SBRC, STOP, FRSCA e SSSPF para todas as cargas simulada. Isto ocorre pois os algoritmos SBPPMC, FIPPMC e MIFMC possuem alta utilização da rede devido ao não uso de agregação de tráfego e sobreposição de espectro nos caminhos de *backup*. Os algoritmos FRSCA e STOP possuem razão de fragmentação semelhantes, apesar do algoritmo FRSCA produzir valores de BBR apenas sob cargas mais altas.

A Figura 5.4 mostra a eficiência energética gerada para a topologia Pan-European. Sob cargas menores, o alto bloqueio de requisições produzido pelo algoritmo SSSPF, juntamente com a utilização de agregação de tráfego e sobreposição de espectro, produz caminhos primários com menor número médio de saltos, garantindo uma maior eficiência energética. No entanto, conforme a carga da rede aumenta, o bloqueio de conexões produzido pelos algoritmos tronam-se mais próximas do bloqueio produzido pelos dos outros algoritmos, conseqüentemente produziram eficiência energética mais próximas. Sob cargas maiores que 175 erlangs, o algoritmo Cap-DPP produz a maior eficiência energética, pois, sob tais cargas, este algoritmo produz os maiores valores de BBR, gerado pela menor razão de fragmentação e o menor número médio de saltos do caminho primário. Os algoritmos SBPPMC, FIPPMC e MIFMC produzem eficiência energética próximas àquelas produzidas pelos algoritmos STOP, SBRC e FRSCA, apesar de não utilizarem agregação de tráfego e sobreposição de espectro.

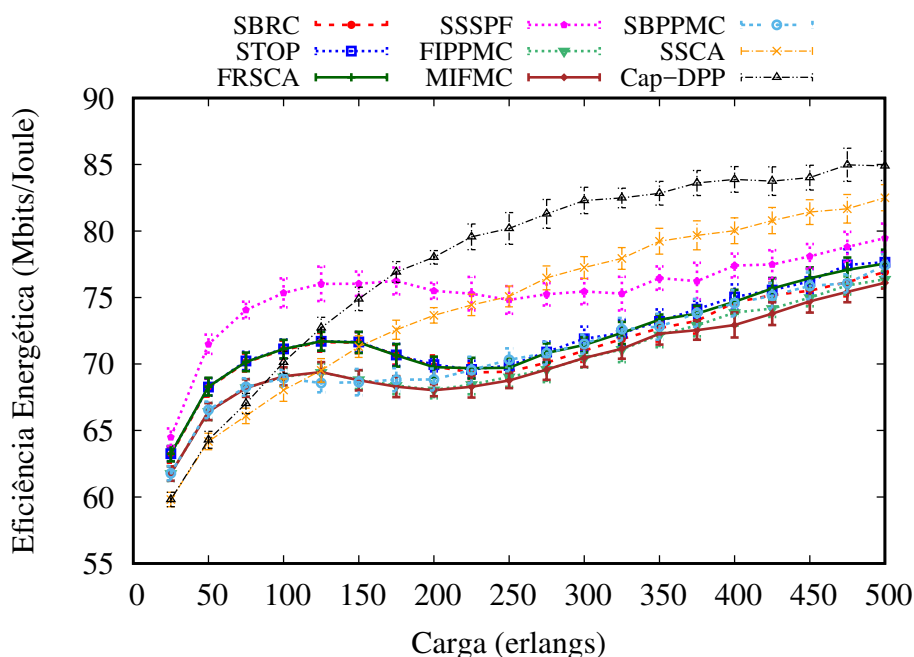


Figura 5.4: Eficiência Energética dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.

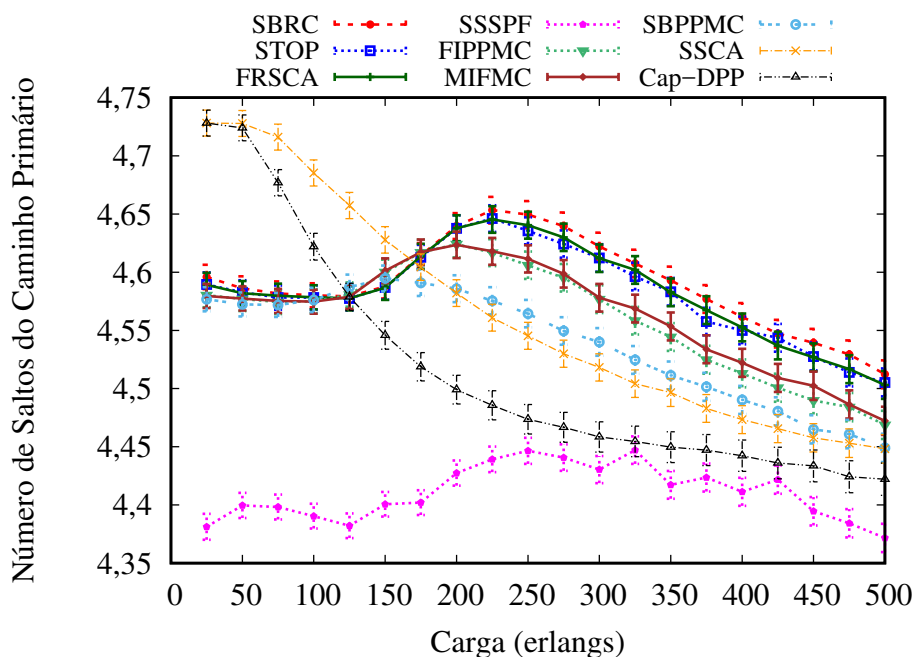


Figura 5.5: Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.

A Figura 5.5 mostra o número médio de saltos do caminho primário gerado para a topologia Pan-European. Sob cargas altas, a utilização de agregação de tráfego e sobreposição de espectro pelos algoritmos SBRC, STOP e FRSCA, para encontrar o menor caminho primário, aumenta a aceitação de requisições, e consequentemente, aumenta o número de saltos dos caminhos

produzidos quando comparado aos algoritmos FIPPMC, SBPPMC e MIFMC. Os algoritmos Cap-DPP e SSCA produzem um maior número médio de saltos do caminho primário para cargas baixas. Isto explica os maiores valores de BBR produzidos. O algoritmo SSSPF possui o menor número médio de saltos do caminho primário, pois este algoritmo produz um alto bloqueio de requisições. A utilização das técnicas de agregação de tráfego e sobreposição de espectro, faz com o algoritmo SSSPF produza uma menor razão de fragmentação. Logo, a BBR, para este algoritmo, é produzida pela indisponibilidade de prover dupla proteção e não pela falta de recursos da rede.

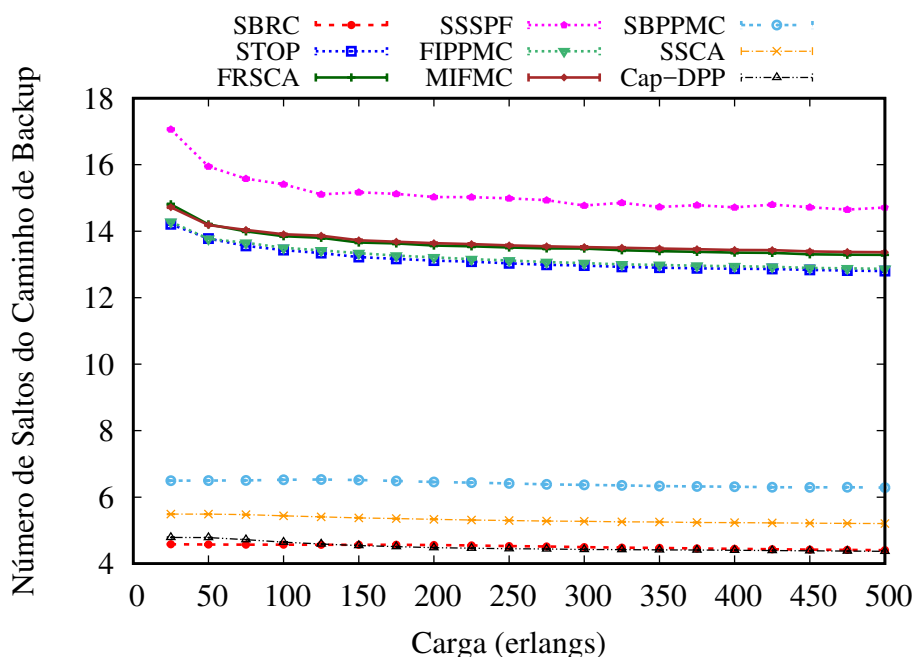


Figura 5.6: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia Pan-European.

A Figura 5.6 mostra o número médio de saltos do caminho de *backup* gerado para a topologia Pan-European. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com maior número médio de saltos que os caminhos que não utilizam. Os algoritmos FIPPMC e MIFMC, que não utilizam agregação de tráfego e sobreposição de espectro, produzem caminhos de *backup* com número médio de saltos, respectivamente, próximos aos algoritmos STOP e FRSCA. O algoritmo FRSCA produz caminhos de *backup* com número médio de saltos sempre maior que o algoritmo STOP, que por sua vez é sempre menor que o algoritmo SSSPF isto ocorre pois o algoritmo SSSPF gera sempre *p-cycles* transzonais ao caminho primário e o algoritmo FRSCA gera sempre que possível esses *p-cycles*. Os *p-cycles* transzonais aos caminhos primário tendem a ser maiores que os caminhos que usam os mesmos enlaces do *p-cycle*. Os algoritmos SBRC e SSCA produzem caminhos de *backup* com número médio de saltos similares, apesar do algoritmo SSCA não utilizar agregação de tráfego e sobreposição de espectro.

A Figura 5.7 mostra a probabilidade de bloqueio para a topologia USA. Enquanto os algoritmos Cap-DPP e o SSCA iniciam o bloqueio de requisições sob cargas de 50 erlangs, os algoritmos FIPPMC e MIFMC iniciam o bloqueio de requisições sob cargas de 125 erlangs e o

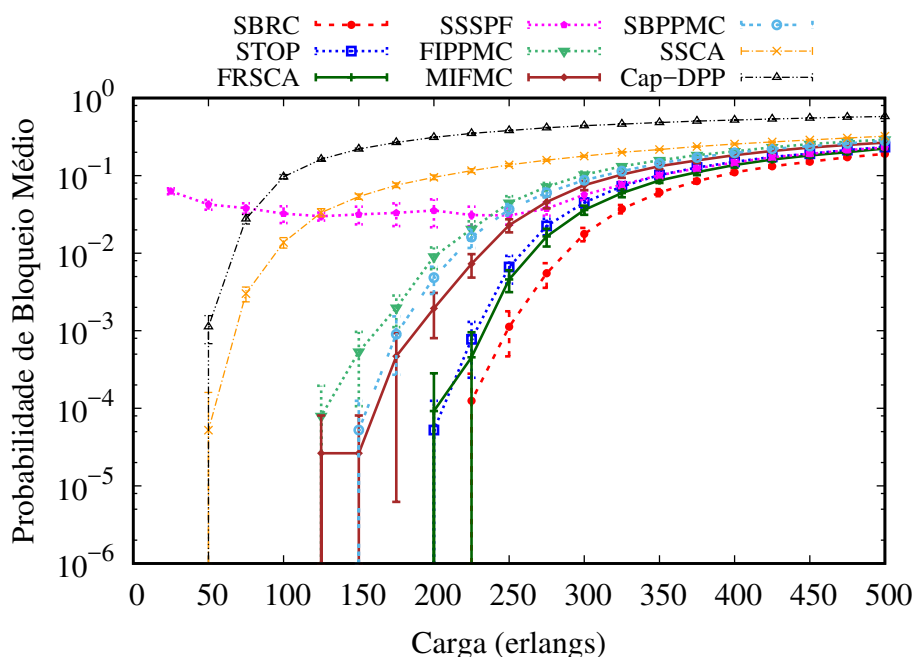


Figura 5.7: Bloqueio de banda dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.

algoritmo SBPPMC inicia o bloqueio de requisições sob carga de 150 erlangs. Por sua vez, os algoritmos FRSCA e STOP iniciam o bloqueio de requisições apenas sob cargas de 200 erlangs e o algoritmo SBRC sob cargas de 225 erlangs. O algoritmo SSSPF inicia o bloqueio de requisições sob cargas de 25 erlangs. O algoritmo SSSPF produz o BBR mais alto, uma vez que ele produz proteção para duas falhas simultâneas. Sob cargas maiores que 125 erlangs, o BBR produzido pelo SSSPF é menor que o produzido pelo algoritmo SSCA, uma vez que o SSCA não usa compartilhamento de espectro e agregação de tráfego. Os algoritmos FRSCA, SBRC e STOP aproveitam a alta conectividade de nós da topologia USA e aumenta a BBR suavemente como uma função do aumento de carga. Para todas as cargas, os algoritmos SBPPMC, FIPPMC e MIFMC que não utilizam agregação de tráfego e sobreposição de espectro, produzem maiores valores de BBR que os algoritmos STOP, SBRC e FRSCA, evidenciando o impacto da utilização de agregação de tráfego e sobreposição de espectro. Sob todas as carga simuladas os algoritmos STOP e FRSCA produzem mais bloqueio que o algoritmo SBRC. O algoritmo SBRC produz a menor BBR para qualquer carga simulada. O baixo BBR produzido por esses algoritmos evidencia a vantagem de utilizar agregação de tráfego e sobreposição de espectro. Além disso, os resultados mostram que o algoritmo SSSPF produz bloqueio aceitável, considerando a topologia simulada e que o algoritmo protege contra duas falhas simultâneas.

A Figura 5.8 mostra a relação de *crosstalk* por *slot* gerado para a topologia USA. O CpS gerado pelos algoritmos FRSCA e SBRC iniciam em 0,05 e aumentam até 0,53 e 0,54, respectivamente. O CpS gerado pelos algoritmos SSSPF e STOP iniciam em 0,04 e 0,06 e aumentam até 0,51, respectivamente. O CpS gerado pelo algoritmo SSCA inicia em 0,01 e aumenta até 0,24, enquanto que o CpS gerado pelo algoritmo Cap-DPP inicia em 0,01 e se mantém quase constante conforme a carga aumenta. O CpS gerado pelos algoritmos FIPPMC e MIFMC iniciam em 0,01 e aumentam até 0,38 e 0,43, respectivamente. Enquanto que o CpS gerado pelo

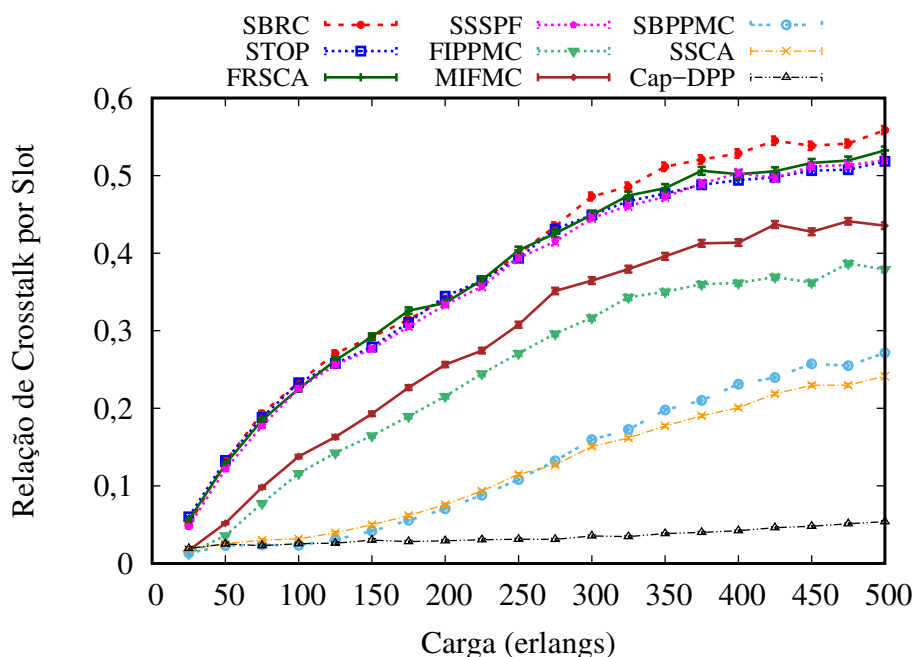


Figura 5.8: *Crosstalk* por *slot* dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.

algoritmo SBPPMC inicia em 0,01 e aumenta até 0,27. O CpS gerado pelos algoritmos que não utilizam agregação de tráfego e sobreposição de espectro são sempre menores do que os algoritmos STOP, SSSPF, SBRC e FRSCA, que utilizam agregação de tráfego e sobreposição de espectro, devido ao menor número de requisições aceitas. O algoritmo Cap-DPP produz os valores CpS mais baixos, como consequência do alto bloqueio e baixa utilização gerada. Os algoritmos que utilizam o multigrafo produzem os valores CpS mais altos, como consequência do baixo bloqueio de requisições gerado e consequentemente da alta utilização. Os algoritmos que utilizam o multigrafo na geração de caminhos obtiveram resultados semelhantes de CpS, como consequência da melhor alocação de recursos proporcionado pelo algoritmo.

A Figura 5.9 mostra a razão de fragmentação gerada para a topologia USA. O algoritmo Cap-DPP produz razão de fragmentação semelhante à dada pelo algoritmo SSCA como consequência da utilização da mesma técnica de caminho utilizada por esses algoritmos. Os algoritmos SBPPMC, FIPPMC e MIFMC que não utilizam agregação de tráfego e sobreposição de espectro produzem maior razão de fragmentação que os algoritmos SBRC, STOP e FRSCA, respectivamente, sob todas cargas. Isto evidencia, as vantagens da agregação de tráfego e sobreposição de espectro em relação a fragmentação da rede. O algoritmo FRSCA possui uma maior razão de fragmentação que o algoritmo STOP, o que explica o menor bloqueio de requisições gerado pelo algoritmo FRSCA. Até 275 erlangs, a menor fragmentação gerada pelo algoritmo SBRC justifica o menor BBR, além disso algoritmos que utilizam *p-cycle* necessitam de mais recursos, então, mesmo com menor razão de fragmentação, as requisições podem produzir maior BBR por falta de recursos. Para cargas baixas, os algoritmos Cap-DPP e SSCA possuem maior razão fragmentação, porém produzem o maior BBR, isso ocorre pois esses algoritmos não utilizam agregação de tráfego e sobreposição de espectro.

A Figura 5.10 mostra a eficiência energética gerada para a topologia USA. Os algoritmos

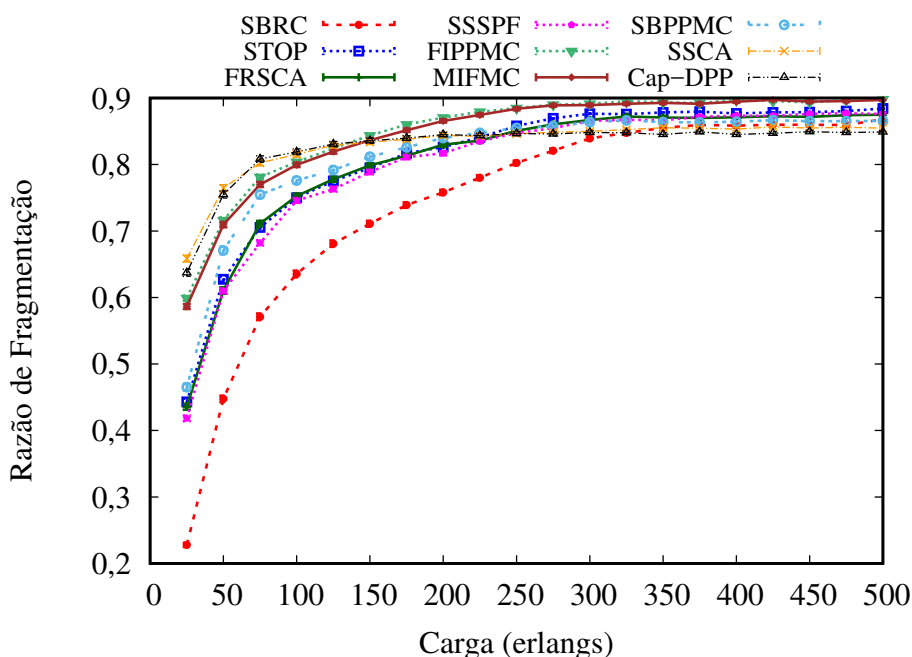


Figura 5.9: Razão de fragmentação dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.

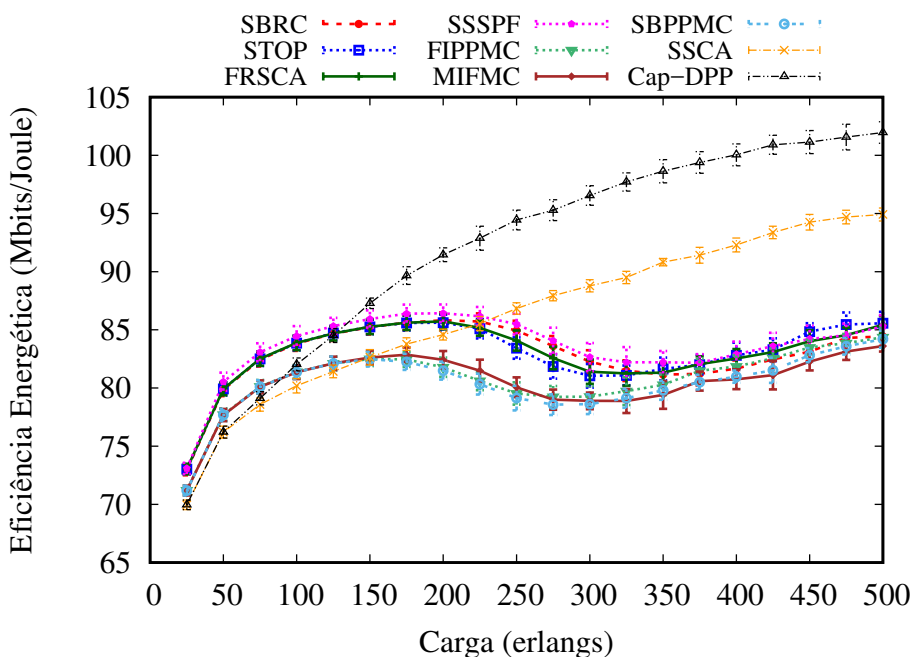


Figura 5.10: Eficiência Energética dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.

que utilizam multigrafo possuem eficiência energética semelhante visto que esses algoritmos utilizam técnicas semelhantes para criação do caminho de *backup*, e caminhos de *backup* que não são utilizados não consomem energia. Os algoritmos FIPPMC, MIFMC e SBPPMC produzem valores de eficiência energética próximos, pois estes algoritmos não utilizam agregação de tráfego e sobreposição e espectro, consequentemente, utilizando um maior número de re-

curso que influencia no consumo energético. Para cargas de até 125 erlangs, os algoritmos que possuem maior número médio de saltos do caminho primário produzem maior BBR e consequentemente menor eficiência energética. Para cargas maiores os algoritmos que utilizam o multigrafo possuem maior número médio de saltos do caminho primário, produzem menor BBR e, consequentemente, menor eficiência energética.

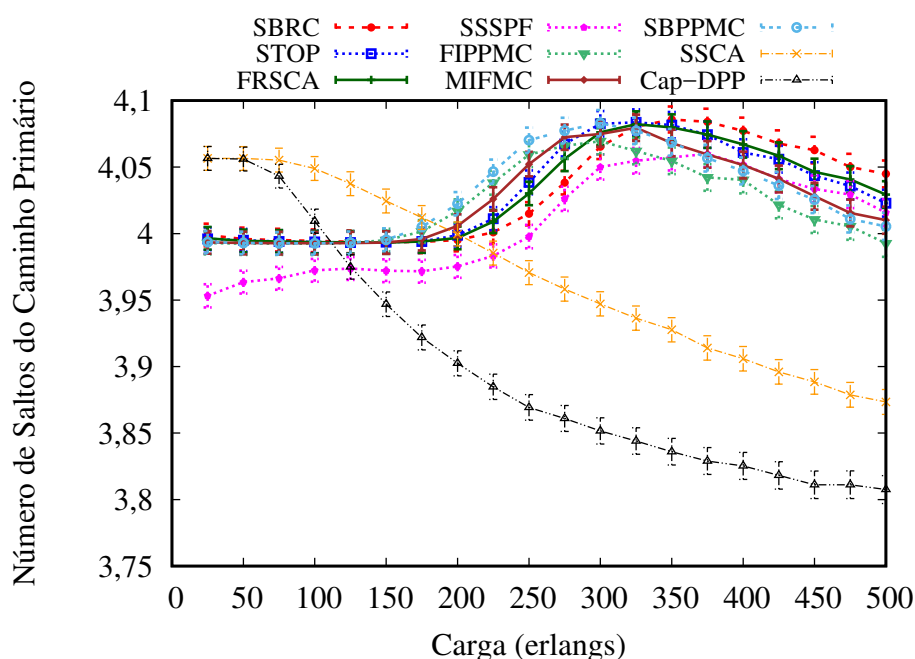


Figura 5.11: Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.

A Figura 5.11 mostra o número médio de saltos do caminho primário gerado para a topologia USA. Os algoritmos SBRC, STOP e FRSCA possuem comportamentos semelhantes para qualquer carga simulada, isto ocorre pois eles utilizam o multigrafo e preocupam-se apenas com proteção contra uma falha. Apesar do algoritmo SSSPF, que realiza proteção contra duas falhas simultâneas, utilizar o multigrafo e possuir comportamento semelhante aos outros algoritmos que também usam, ele possui sempre número médio de saltos do caminho primário menor que os dos algoritmos SBRC, STOP e FRSCA. Isto ocorre pois os maiores valores de BBR produzido pelo algoritmo SSSPF com relação aos outros algoritmos é consequência da inexistência da possibilidade de criação de caminhos de *backup*, deixando mais recursos disponíveis. Sob cargas baixas os algoritmos SBPPMC, FIPPMC e MIFMC produzem caminhos semelhantes aos produzidos pelos algoritmos SBRC, STOP e FRSCA. Isto acontece, pois sob estas cargas os algoritmos que não utilizam agregação de tráfego e sobreposição de espectro não produzem bloqueio de requisições. Os maiores valores de BBR produzidos pelos algoritmos Cap-DPP e SSCA faz com que esses algoritmos gerem caminhos menores, visto que, quando se tem maiores valores de BBR, alguns enlaces da rede são sobrecarregados, impossibilitando caminhos maiores.

A Figura 5.12 mostra o número médio de saltos do caminho de *backup* gerado para a topologia USA. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com três vezes

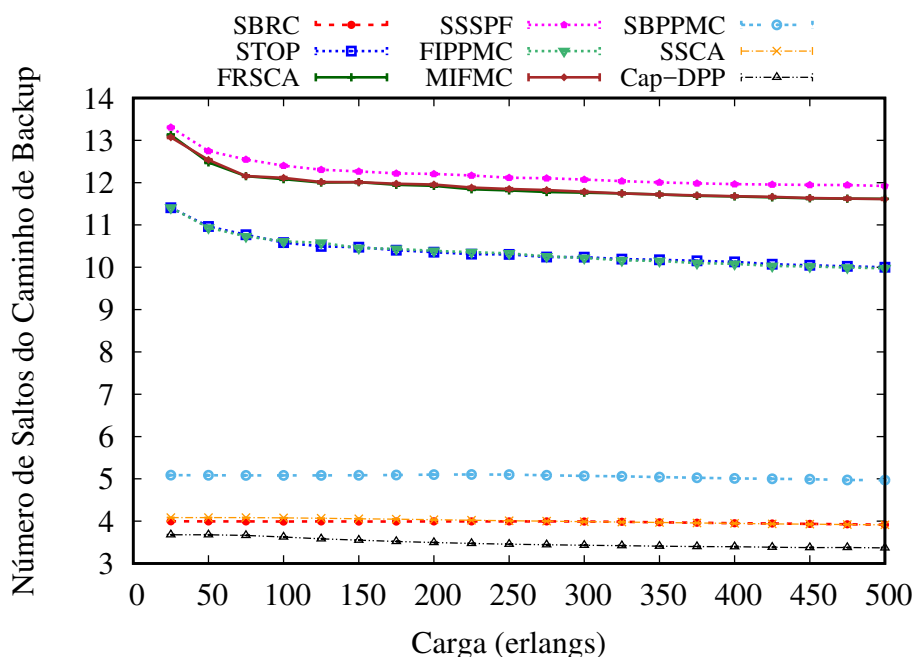


Figura 5.12: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia USA.

mais saltos do que os caminhos que não o utilizam. O algoritmo SSSPF, que só admite caminhos de *backup* que protejam contra duas falhas simultâneas, possui o maior número médio de saltos do caminho de *backup*. Isto ocorre pois ele só admite caminhos protegidos por p -cycles transzonais ao caminho primário, e esses p -cycles tendem a ser maiores que os caminhos que usam os mesmos enlaces do p -cycle. O algoritmo FRSCA possui o segundo maior número médio de saltos do caminho de *backup*. Isto ocorre pois este algoritmo prioriza os caminhos transzonais ao caminho primário, mas permite a criação de caminhos que não o sejam. Por sua vez, dentre os algoritmos que utilizam p -cycle, o algoritmo STOP é o que possui o menor número médio de saltos do caminho de *backup*. Isto ocorre pois este algoritmo procura utilizar o menor p -cycle para realizar a proteção do caminho. O algoritmo Cap-DPP produz o menor número de saltos para o caminho de *backup*, pois esse algoritmo produz maiores valores de BBR. Os algoritmos SBRC e SSCA produzem caminhos de *backup* com número médio de saltos similar, como consequência de ambos utilizarem a técnica SBPP. O algoritmo STOP que utiliza agregação de tráfego e sobreposição de espectro produz número de saltos dos caminhos de *backup* próximos que aqueles produzidos pelo algoritmo FIPPMC. Por sua vez, o algoritmo FRSCA que também utiliza agregação de tráfego e sobreposição de espectro produz número de saltos dos caminhos de *backup* próximos que aqueles produzidos pelo algoritmo MIFMC. Por outro lado, o algoritmo SBRC, que utiliza agregação de tráfego e sobreposição de espectro, produz número de saltos dos caminhos de *backup* menor que aqueles produzidos pelo algoritmo SBPPMC.

A Figura 5.13 mostra a probabilidade de bloqueio para a topologia NSF. O baixo grau dos nós nesta topologia leva à criação de gargalos e a um aumento muito mais rápido nos valores de BBRs quando comparado com os valores de BBRs produzidos pelas outras topologias. En-

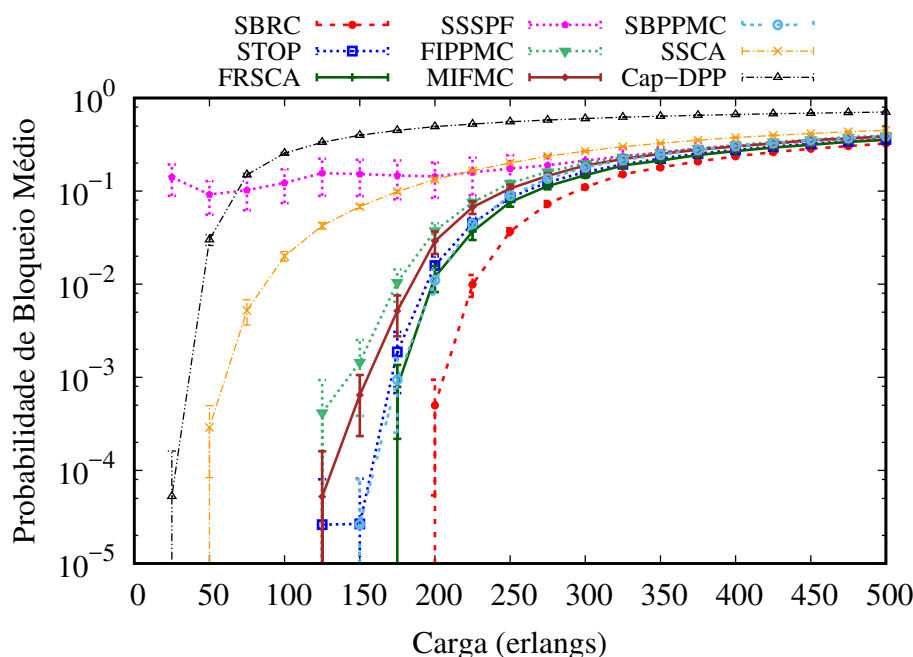


Figura 5.13: Bloqueio de banda dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.

quanto os algoritmos FIPPMC e MIFMC iniciam o bloqueio de requisições sob cargas de 125 erlangs. O algoritmo SBPPMC inicia o bloqueio de requisições apenas sob cargas de 175 erlangs. Por sua vez, enquanto os algoritmos Cap-DPP e SSCA iniciam o bloqueio de requisições sob carga de 25 e 50 erlangs, respectivamente, o algoritmo STOP inicia o bloqueio apenas sob cargas de 125 erlangs. Enquanto o algoritmo FRSCA inicia o bloqueio de requisições sob 175 erlangs, o algoritmo SBRC inicia o bloqueio apenas sob 200 erlangs. Além disso, sob todas as cargas simuladas o algoritmo SBRC produz valores menores de BBR que o produzido pelo algoritmo FRSCA. O algoritmo SBRC produz valores menores de BBR pois a geração de *p-cycle* é mais difícil para as topologias com menor conectividade. Com exceção do algoritmo SSSPF, que realiza proteção contra duas falhas simultâneas, os algoritmos que utilizam agregação de tráfego e sobreposição de espectro produzem menor BBR. Sob carga de 200 erlangs o algoritmo SBRC produz valores de BBR uma ordem de grandeza menor que a dos algoritmos FRSCA e STOP. Sob cargas maiores, todos os algoritmos que compartilham o caminho de *backup* produzem valores de BBRs semelhantes.

A Figura 5.14 mostra a relação de *crossstalk* por *slot* gerado para a topologia NSF. O CpS gerado pelo algoritmo SSSPF inicia com um valor de 0,07 e aumenta até 0,64. Os algoritmos FRSCA, STOP e SBRC possuem os maiores valores de CpS. Isto acontece devido aos valores menores de BBR gerados por estes algoritmos. Os valores de CpS dos algoritmos SBPPMC, FIPPMC e MIFMC iniciam em 0,02 e aumentam até 0,50, 0,57 e 0,58, respectivamente. Os valores de CpS dos algoritmos STOP, FRSCA e SBRC são sempre mais altos do que aqueles produzidos pelos algoritmos SBPPMC, FIPPMC e MIFMC. Isto acontece pela não utilização de agregação de tráfego e sobreposição de espectro pelos algoritmos SBPPMC, FIPPMC e MIFMC. O alto CpS gerado pelo algoritmo SSSPF, mesmo com altos valores de BBR, ocorre devido a necessidade de proteção contra duas falhas simultâneas, o que faz com que o algo-

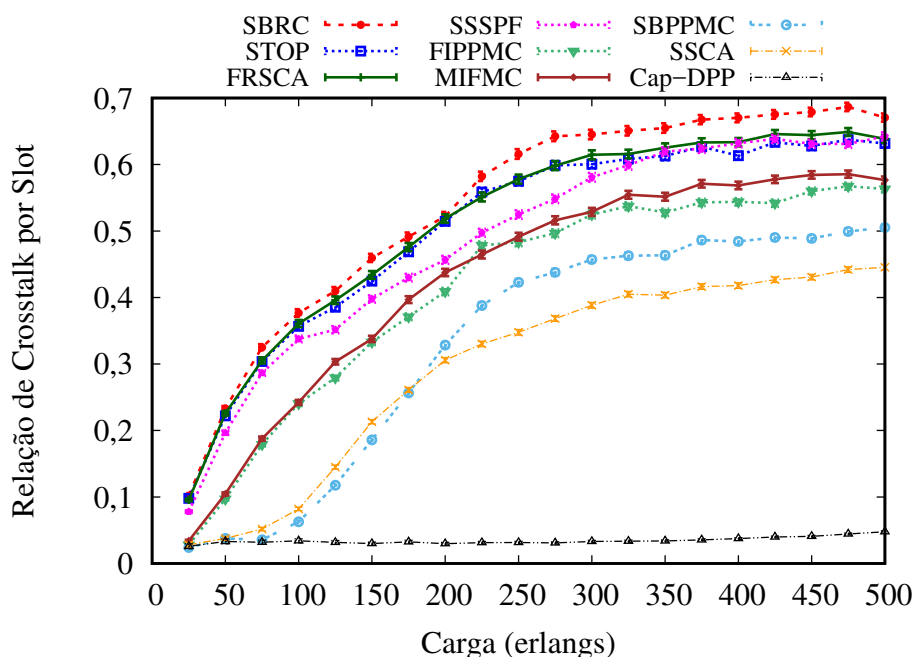


Figura 5.14: *Crosstalk* por *slot* dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.

ritmo utilize mais recursos da rede. Os algoritmos Cap-DPP e SSCA produzem os valores de CpS mais baixos quando comparados aos valores CpS gerados pelos outros algoritmos, como consequência do alto BBR produzido e da baixa utilização.

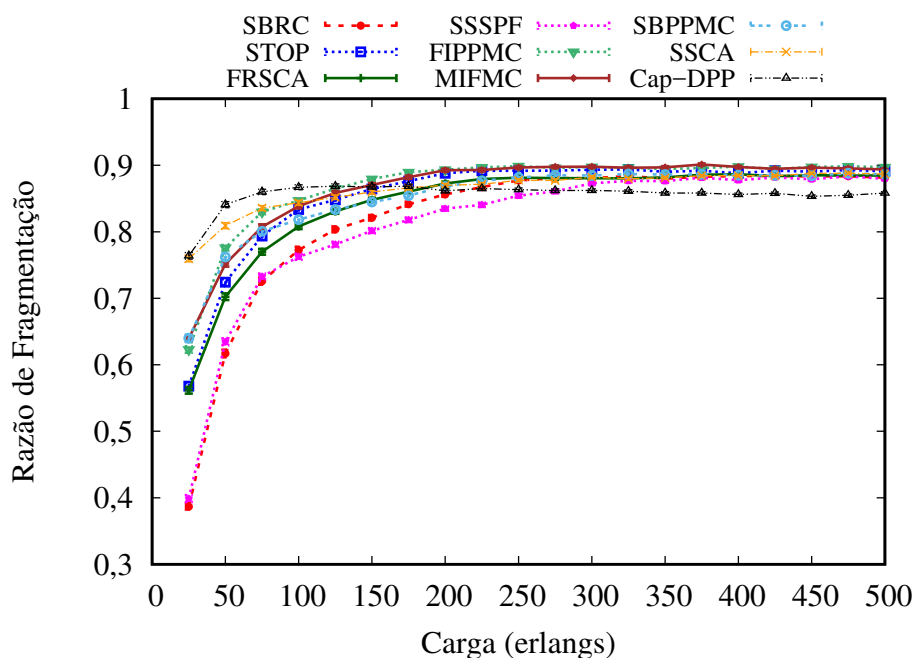


Figura 5.15: Razão de fragmentação dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.

A Figura 5.15 mostra a razão de fragmentação gerada para a topologia NSF. O algoritmo SSSPF possui a menor razão de fragmentação quando comparado a razão de fragmentação pro-

duzida pelos outros algoritmos simulados. Isto acontece pois a criação de proteção contra dupla falhas simultâneas exige mais recursos, logo a rede produz maior fragmentação. Sob baixas cargas, os algoritmos SBPPMC, FIPPMC e MIFMC produzem maiores valores de fragmentação do que os algoritmos SBRC, STOP e FRSCA. Isto justifica o alto bloqueio produzido pelos algoritmos SBPPMC, FIPPMC e MIFMC, uma vez que a maior fragmentação dificulta a alocação de novos caminhos. Além disso, a menor razão de fragmentação dos algoritmos SBRC, STOP e FRSCA é consequência da utilização de agregação de tráfego e sobreposição de espectro. Para cargas menores que 125 erlangs, o algoritmo Cap-DPP produz a maior razão de fragmentação, apesar deste algoritmo produzir maior BBR, pois os algoritmos que utilizam agregação de tráfego e sobreposição de espectro são mais eficientes que os que não utilizam. Conforme os outros algoritmos aumentam, a BBR produzida pelo algoritmo Cap-DPP mantém sua razão de fragmentação. Além disso, sob altas cargas, este algoritmo produz a menor razão de fragmentação.

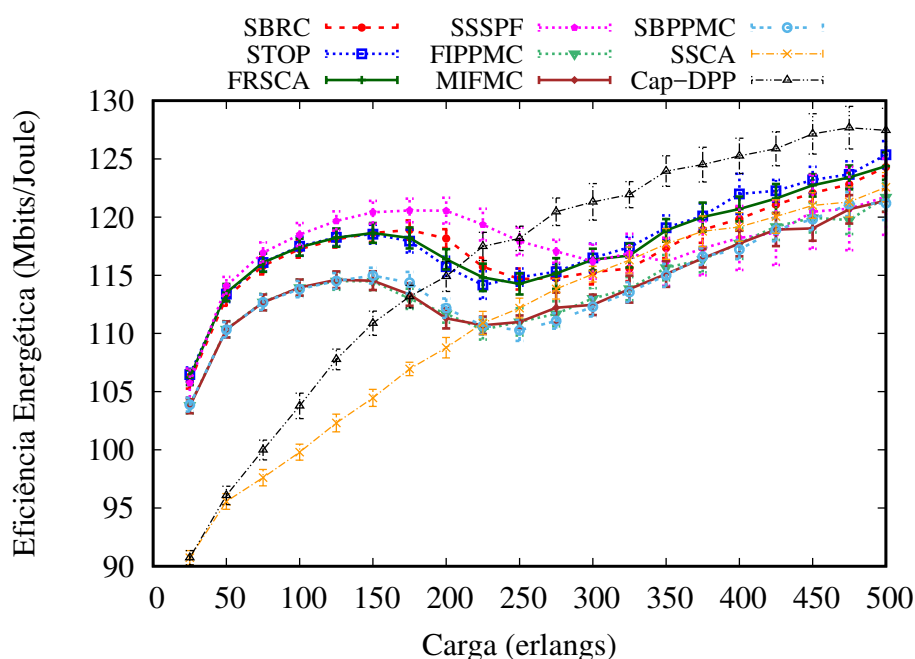


Figura 5.16: Eficiência Energética dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.

A Figura 5.16 mostra a eficiência energética gerada para a topologia NSF. Sob baixas cargas, a eficiência energética produzida pelo SSSPF é maior que a produzida pelos outros algoritmos, pois, além deste algoritmo utilizar sobreposição de espectro e agregação de tráfego, ele produz um alto BBR gerado por não conseguir satisfazer a restrição de dupla proteção contra falhas simultâneas. Não há muita diferença entre a eficiência energética produzida pelos algoritmos SBRC, FRSCA e STOP. Isto reflete o número médio de saltos do caminho primário que eles possuem. Os algoritmos STOP, FRSCA e SBRC produzem maior eficiência energética do que os algoritmos SBPPMC, FIPPMC e MIFMC que não utilizam agregação de tráfego e sobreposição de espectro, consequentemente, utilizando mais recursos o que influencia no consumo energético. Até carga de 325 erlangs, o algoritmo SSCA possui a menor eficiência energética, pois este algoritmo aloca o maior número de saltos para o caminho primário.

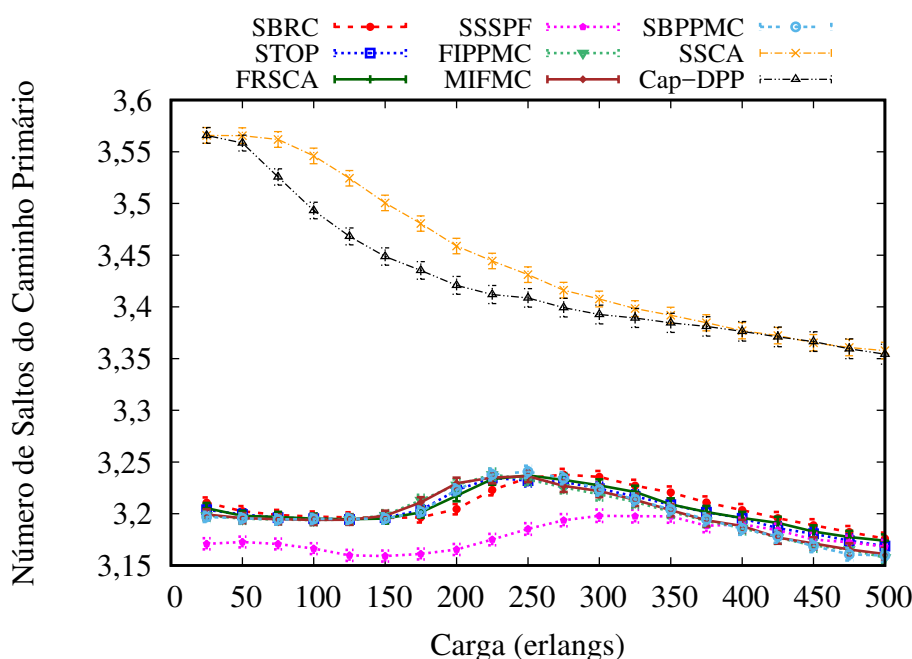


Figura 5.17: Número médio de saltos do caminho primário dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.

A Figura 5.17 mostra o número médio de saltos do caminho primário gerado para a topologia NSF. Os algoritmos que utilizam o multigrafo possuem sempre menor número médio de saltos que os caminhos que não utilizam o multigrafo. Os algoritmos Cap-DPP e SSCA alocam caminhos primários com número médio de saltos semelhantes quando comparados com outros algoritmos. Os algoritmos SBRC, STOP, FRSCA, FIPPMC, SBPPMC e MIFMC produzem caminhos primários com número médio de saltos menores, principalmente sob cargas baixas. Isto acontece devido a similitude na criação dos caminhos primários e da proximidade do número de requisições aceitas sob estas cargas. Como nas topologias anteriores, os algoritmos STOP, SBRC e FRSCA que utilizam o multigrafo alocam caminhos primários com número médio de saltos semelhantes sob qualquer carga. Apesar do algoritmo SSSPF utilizar o multigrafo, agregação de tráfego e sobreposição de espectro, devido a maior BBR produzido com a necessidade de produzir dupla proteção, este algoritmo produziu um menor número médio de saltos para os caminhos primários, principalmente para cargas mais baixas.

A Figura 5.18 mostra o número médio de saltos do caminho de *backup* gerado para a topologia NSF. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com duas vezes mais número médio de saltos que os caminhos que não utilizam. A menor conectividade da topologia NSF influenciou os tamanhos dos *p-cycles* criados de forma que o algoritmo SSSPF produziu maior número médio de saltos que o FRSCA, que, por sua vez, possui maior número médio de saltos que o STOP. Os algoritmos SBPPMC, FIPPMC e MIFMC que não utilizam agregação de tráfego e sobreposição de espectro produzem, respectivamente, caminhos de *backup* com maior número médio de saltos do que os algoritmos SBRC, STOP e FRSCA. Isto ocorre devido a menor utilização de recursos quando utilizada as técnicas de agregação de tráfego e sobreposição de espectro. O número de saltos do caminho de *backup* para os algoritmos

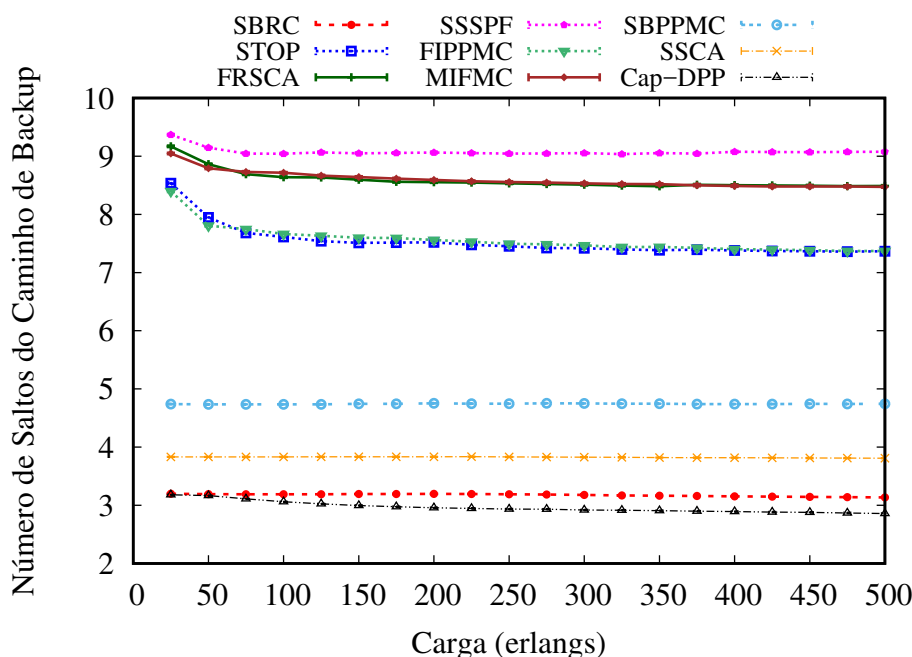


Figura 5.18: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPP, SSCA, FIPPMC, SBPPMC, MIFMC, STOP, SSSPF, SBRC e FRSCA em função da carga da rede para topologia NSF.

Cap-DPP e SBRC são semelhantes, o que sugere uma melhor alocação de recurso pelo algoritmo SBRC, visto que este produz um menor bloqueio de requisições.

5.8 Síntese do capítulo

Neste capítulo, apresentaram-se os algoritmos STOP, SBRC, SSSPF e FRSCA para proteção de redes ópticas elásticas com multiplexação espacial, utilizando agregação de tráfego e sobreposição de espectro. Como nos capítulos anteriores, os algoritmos apresentados utilizaram o multigrafo para encontrar caminhos através de topologias com múltiplos núcleos.

Os algoritmos apresentados são extensões dos algoritmos do Capítulo 3, adicionando agregação de tráfego e sobreposição de espectro para os caminhos de *backup*. Além disto, um algoritmo para prover proteção contra dupla falhas simultâneas é proposto. Os resultados mostraram uma grande redução no BBR produzida para os algoritmos propostos neste capítulo com relação aos apresentados no Capítulo 3.

Os valores de BBR produzidos pelos algoritmos propostos evidenciam o benefício de utilizar agregação de tráfego e sobreposição de espectro. Além disso, o algoritmo para dupla falha, mostrou-se bastante aceitável, principalmente para cargas altas, quando considera-se que as topologias simuladas não estão preparadas para fornecer dupla proteção.

Os algoritmos que utilizam agregação de tráfego e sobreposição de espectro obtiveram baixa razão de fragmentação, porém a BBR produzida por estes algoritmos também reduziram. Mesmo utilizando agregação de tráfego e sobreposição de espectro, o CpS produzido por esses algoritmos continuou alto, devido à alta utilização dos recursos da rede.

Capítulo 6

Modulação adaptativa, agregação de tráfego e sobreposição de espectro para proteção de redes ópticas elásticas com multiplexação espacial

Neste capítulo, avalia-se o impacto de três novos algoritmos para proteção em redes ópticas elásticas com multiplexação espacial utilizando modulação adaptativa, agregação de tráfego e sobreposição de espectro. Os algoritmos apresentados estendem os algoritmos apresentados no capítulo 5, adicionando aos algoritmos a modulação adaptativa. Os três algoritmos apresentados neste capítulo diferem entre si apenas na forma de proteção realizada. O uso de modulação adaptativa junto com agregação de tráfego e sobreposição de espectro, diminui o número de *slots* utilizados nos caminhos ópticos. Enquanto a modulação adaptativa possibilita o envio de uma maior quantidade de bits por símbolos, a agregação de tráfego possibilita a não utilização de banda de guarda entre caminhos.

Neste capítulo, serão utilizados os valores apresentados na tabela 4.1. O objetivo dos algoritmos é reduzir a probabilidade de bloqueio e garantir proteção das requisições de estabelecimento de conexões e evitar a degradação da qualidade da transmissão. Os algoritmos propostos definem duas rotas do nó de origem para o nó de destino, selecionando os formatos de modulação para a distância entre os nós origem e destino. Além disso, escolhem uma faixa de espectro livre (isto é, um conjunto de *slots* de frequências contíguas) na rota definida para estabelecer o circuito óptico. A qualidade dos algoritmos propostos determinam o número de circuitos que podem ser estabelecidos (e também bloqueados) em uma determinada topologia de rede.

Para avaliar a efetividade dos algoritmos propostos, foram realizadas simulações utilizando as três topologias apresentadas no capítulo 3. Os resultados obtidos mostram-se satisfatórios, além do que, os algoritmos propostos geram menos bloqueio do que os outros algoritmos apresentados nos capítulos anteriores. Além disso, os algoritmos propostos possuem diferentes comportamentos, associados à topologia utilizada.

Este capítulo está organizado conforme segue. Na Seção 6.1, apresenta-se uma revisão bibliográfica. Na seção 6.2 expõe-se a notação utilizada nos algoritmos. Na Seção 6.3, introduz-se o algoritmo RAMBO. Na Seção 6.4, apresenta-se o algoritmo APPROVE. Na Seção 6.5, expõe-se o algoritmo MAROTO. Na Seção 6.6, compara-se o desempenho dos algoritmos propostos

com outros algoritmos. Finalmente, na Seção 6.7, apresenta-se a síntese do capítulo.

6.1 Trabalhos relacionados

Na literatura, poucos trabalhos estudam agregação de tráfego e modulação adaptativa em conjunto. Além disso, nenhum trabalho estuda agregação de tráfego e modulação adaptativa para proteção de redes ópticas elásticas utilizando multiplexação espacial.

Os autores em [99] avaliaram o efeito do uso da modulação adaptativa para a redução dos custos em redes ópticas elásticas que empreguem modulação adaptativa e agregação de tráfego. O trabalho mostra as vantagens da modulação adaptativa usando exemplos ilustrativos e uma avaliação de custo definida no artigo.

Em [101], foi proposto um algoritmo heurístico para agregação de tráfego em redes ópticas elásticas, que considera a modulação adaptativa. O algoritmo produz uma topologia virtual que consiste em demandas de caminho óptico e rotas de fluxos de tráfego assumindo a condição de tráfego estático. A avaliação mostrou que o algoritmo economiza significativamente caminhos ópticos em comparação com outros algoritmos em várias cenários de rede.

Propuseram em [114] um algoritmo que considera modulação adaptativa e agregação de tráfego em redes ópticas elásticas com baixo consumo energético. Utiliza-se um algoritmo para agregação de tráfego com baixo consumo energético, previamente proposto pelos autores. As simulações mostraram que a modulação adaptativa juntamente com agregação de tráfego com baixo consumo energético desempenham um papel importante na economia de recursos e consumo de energia para rede EONs, uma vez que a largura de banda espectral melhora conforme aumenta-se o número de bits por símbolo para transmitir a mesma taxa de dados aumenta.

Em [7], os autores estenderam algoritmos de agregação de tráfego em redes ópticas elásticas adicionando uso de diferentes formatos de modulação (usados individualmente ou em conjunto), bem como avaliaram o desempenho dos métodos quando existem restrições de comprometimento da camada física.

Tabela 6.1: Comparação dos trabalhos relacionados com os algoritmos RAMBO, APPROVE e MAROTO.

Abordagem	Proteção	SDM	TG	SE	<i>p-Cycle</i>	SBPP	AM
Takita <i>et al.</i> [99]	Não	Não	Sim	Não	Não	Não	Sim
Tanaka <i>et al.</i> [101]	Não	Não	Sim	Não	Não	Não	Sim
Cunqian <i>et al.</i> [114]	Não	Não	Sim	Não	Não	Não	Sim
Castro <i>et al.</i> [7]	Não	Não	Sim	Não	Não	Não	Sim
RAMBO	Sim	Sim	Sim	Sim	Não	Sim	Sim
APPROVE	Sim	Sim	Sim	Sim	Sim	Não	Sim
MAROTO	Sim	Sim	Sim	Sim	Sim	Não	Sim

Apesar das técnicas de agregação de tráfego e modulação adaptativa já terem sido estudadas para redes ópticas elásticas, elas não foram estudadas em redes óptica elásticas com multiplexação espacial. Além disso, estas técnicas não foram estudadas em conjunto com a técnica de sobreposição de espectro.

A Tabela 6.1 apresenta uma comparação entre os trabalhos relacionados e os algoritmos propostos neste capítulo. Os trabalhos foram categorizados com relação à proteção, ao uso de multiplexação espacial, agregação de tráfego (TG), sobreposição de espectro (SE), *p-cycle* para fornecer proteção, a utilização caminhos de *backup* compartilhados e, por fim, modulação adaptativa (AM).

6.2 Notação

A notação matemática a seguir será usada neste capítulo. A notação incrementa as notações apresentadas nos Capítulos 3, 4 e 5. As principais diferenças encontram-se nos aspectos que tratam agregação de tráfego, sobreposição de espectro e modulação adaptativa em conjunto:

$\widetilde{G}'_{n,b_m} = (\widetilde{V}, \widetilde{E}, \widetilde{W})$: o n -ésimo grafo tal que \widetilde{E} é o conjunto de arestas conectando $\{\widetilde{u}, \widetilde{v}\} \in \widetilde{V}$ e \widetilde{W} é conjunto de custos associados a \widetilde{E} . As arestas em \widetilde{E} correspondem ao mapeamento de b_m arestas em G iniciando na n -ésima aresta;

$\Delta(G, C, b_m) = \{\widetilde{G}'_{n,b_m}\}$: função que produz todos os σ grafos de G considerando as possíveis agregações de tráfego e cada aresta dos grafos gerados representam b_m slots;

$\widetilde{G}''_{n,b_m} = (\widetilde{V}, \widetilde{E}, \widetilde{W})$: o n -ésimo grafo tal que \widetilde{E} é o conjunto de arestas conectando $\{\widetilde{u}, \widetilde{v}\} \in \widetilde{V}$ e \widetilde{W} é conjunto de custos associados a \widetilde{E} . As arestas em \widetilde{E} correspondem ao mapeamento de b_m arestas em G iniciando na n -ésima aresta;

$\kappa(G, C, b_m, P_n) = \{\widetilde{G}'_{n,b_m}\}$: função que produz todos os grafos de G , considerando que os *slots* de proteção podem ser sobrepostos, desde que os caminhos de trabalho (P_n) que são protegidos pelos caminhos de *backup* e que usam o *slot* sejam fisicamente disjuntas (sobreposição de espectro). Cada aresta dos grafos gerados representam b_m slots;

6.3 O algoritmo RAMBO

O algoritmo *Routing, Adaptive Modulation, Backuping path and spectrum Overlap* (RAMBO) procura o formato de modulação apropriado para uma determinada requisição, a partir dos níveis mais altos de modulação. Além disso, este algoritmo pode utilizar agregação de tráfego na criação dos caminhos primários e nos caminhos de *backup*. Adicionalmente, ele pode utilizar sobreposição de espectro na formação dos caminhos de *backup*. No algoritmo 12, não só os caminhos de *backup* podem ser compartilhados como os *slots* podem ser sobrepostos. O formato de modulação é selecionado para manter a qualidade da transmissão (QoT).

No algoritmo RAMBO (Algoritmo 12), a complexidade de transformar multígrafos em grafos aumenta, visto que, conforme a modulação é alterada, há a necessidade de criação de novos grafos. Além disso, a criação dos grafos considera a possível agregação de tráfego. Para uma dada requisição, o algoritmo escolhe a modulação com a maior eficiência espectral e transforma o multigrafo em $C \times (N - b_m + 1)$ grafos (linha 2), a criação dos grafos depende exclusivamente da banda requerida e da capacidade do *slot* utilizando determinado formato de modulação. Após a geração dos grafos, o algoritmo utiliza o algoritmo de Dijkstra para encontrar o menor caminho (linha 3). A linha 4 escolhe o menor caminho e o peso do caminho dentre os menores caminhos dos grafos. Após escolher o caminho considerando os grafos gerados para diferentes formatos de modulação as correspondentes arestas do caminho têm seus pesos

Algoritmo 12 RAMBO**Entrada:** $G, r(s, d, b)$ **Saída:** Caminho primário e Caminho de backup

```

1: Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
2:    $\Delta(G, C, b_m)$ 
3:    $(W(P_n), P_n) = \alpha(\widetilde{G}'_{n,b_m}, r(s, d, b)) \quad \forall n \in \sigma$ 
4:    $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
5: Fim Para todo
6:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
7: Se  $W_{P_{s,d}} = \infty$  então
8:   bloqueia requisição  $(r(s, d, b))$ 
9: Senão
10:  Se  $\exists \Omega(P_n, Z, r(s, d, b))$  então
11:    estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
12:  Senão
13:    Para todo  $m \in M \wedge W_{Z_{s,d}} = \infty$  faça
14:       $\kappa(G, C, b_m, P_n)$ 
15:       $(W(Z_n), Z_n) = \Phi(\widetilde{G}''_{n,b_m}, P_n, r(s, d, b)) \quad \forall n$ 
16:       $W_{Z_{s,d}} = W(Z_n) \mid \forall i W(Z_n) \leq W(Z_i)$ 
17:    Fim Para todo
18:    Se  $W_{Z_{s,d}} = \infty$  então
19:      bloqueia requisição  $(r(s, d, b))$ 
20:    Senão
21:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
22:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in Z_i$ 
23:    Fim Se
24:  Fim Se
25: Fim Se

```

configurados para ∞ (linha 6). A linha 1 escolhe o melhor formato de modulação para gerar o menor caminho. Se não for possível encontrar um caminho que respeite as restrições de contiguidade e continuidade para a demanda (linha 7), a conexão é bloqueada (linha 8). Caso contrário, busca-se um caminho de *backup* dentre os caminhos de *backup* já estabelecidos para outras requisições. Para que um caminho de *backup* seja válido, os *slots* que fazem parte do caminho primário (H_{P_n}) precisam ser disjuntos de todos os *slots* dos caminhos já protegidos ($H_{P_{Z_n}}$) pelo caminho de backup (Z_n). Se existir um caminho de *backup* disponível (linha 10), a requisição é estabelecida (linha 11). Se não existir nenhum caminho de *backup* que possa proteger a requisição, então, um novo caminho precisa ser criado. A criação do caminho de *backup* pode utilizar formato de modulação diferente do caminho primário. Logo, o processo de transformação do multigrafo em grafos necessita ser refeito (linha 14). No entanto, a criação dos grafos considera não somente agregação de tráfego como também o compartilhamento de espectro. As linhas 15 e 16 buscam o menor caminho de *backup* disjunto do caminho primário para o formato de modulação. Caso nenhum caminho de *backup* possa ser criado, a requisição é bloqueada (linha 19). Caso contrário, a requisição é aceita, os caminhos são estabelecidos e as correspondentes arestas do multigrafo G tem seus pesos alterados para ∞ (linha 22).

A complexidade do algoritmo RAMBO é analisada como segue. No pior caso, a modulação

escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $M \times (E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b_m)$ grafos, onde a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, no pior caso para o caminho de backup, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b_m)$ grafos. Logo, no pior caso, o algoritmo RAMBO tem complexidade de $2 \times (M \times (E + V) + C \times (N - b_m) \times (E + V \log V))$. Como M , N , C e b_m podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

6.4 O algoritmo APPROVE

Como no algoritmo 12, o algoritmo *Adaptive modulation, P-cycle Path, Routing and spectrum OVERlap* (APPROVE) procura o formato de modulação apropriado para uma determinada requisição, a partir dos níveis mais baixos de modulação. Além disso, este algoritmo utiliza agregação de tráfego e sobreposição de espectro nos caminhos de *backup*. Contudo, uma importante diferença entre o algoritmo APPROVE e o algoritmo RAMBO está na criação do caminho de backup, que ao invés de utilizar a técnica SBPP utiliza *p-cycles*. Diminuindo o tempo de comutação de proteção em relação a técnica SBPP e provendo proteção a um número maior de caminhos com apenas um *p-cycle*.

No algoritmo APPROVE, a linha 1 escolhe o formato de modulação a ser utilizado pelo menor caminho. A linha 2 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação e na possível utilização da agregação de tráfego. As linhas 3 e 4 computam o menor caminho e o custo a partir dos grafos gerados. O processo de escolha do formato de modulação (linha 1) é repetido até que um caminho seja encontrado ou não exista mais formato de modulação viável para transmissão. Na linha 6, o peso das arestas do caminho escolhido é marcado com ∞ . Se nenhum caminho primário for encontrado (linha 7), a requisição é bloqueada (linha 8). Na linha 10, é buscado um *p-cycle* dentre os *p-cycles* ativos que possa proteger a requisição. Para a escolha do *p-cycle*, é levado em consideração que todos os caminhos primários protegidos por um *p-cycle* (P_{T_n}) devem ser disjuntos, os nós finais do caminho primário precisam estar no *p-cycle*, e o possível caminho a ser utilizado pela requisição precisa respeitar a eficiência espectral. Além disso, os slots que fazem parte do caminho primário (H_{T_n}) precisam ser disjuntos de todos os slots dos caminhos já protegidos ($H_{P_{T_n}}$) pelo *p-cycle* (T_n). Com a utilização de *p-cycle* e modulação, surge o problema de que a eficiência espectral deve levar em conta a distância máxima da fibra utilizada e não a distância de percorrer todo o *p-cycle*. Em um caso em que o caminho primário está no *p-cycle*, a distância a ser levada em consideração é a dos enlaces que não fazem parte do caminho. Por outro lado, para o caminho primário que é transzonal ao *p-cycle* a distância considerada será a menor distância entre os nós de origem e destino. Caso exista um *p-cycle* que satisfaça os requisitos para proteger a requisição, a conexão é estabelecida (linha 10). Caso contrário, um *p-cycle* para proteger o caminho primário deve ser criado. A criação do *p-cycle* precisa manter os requisitos mínimos de QoT. Para criação do *p-cycle*, escolhe-se um formato de modulação (linha 13). Na criação do *p-cycle*, a agregação de tráfego e sobreposição de espectro também será considerada. São gerados $C \times (N - b_m + 1)$ grafos a partir do multigrafo (linha 14). As

Algoritmo 13 APPROVE**Entrada:** $G, r(s, d, b)$ **Saída:** Caminho primário e Caminho de backup

```

1: Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
2:    $\Delta(G, C, b_m)$ 
3:    $(W(P_n), P_n) = \alpha(\tilde{G}_{n,b_m}, r(s, d, b)) \quad \forall n \in \sigma$ 
4:    $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
5: Fim Para todo
6:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
7: Se  $W_{P_{s,d}} = \infty$  então
8:   bloqueia requisição  $(r(s, d, b))$ 
9: Senão
10:  Se  $\exists v(P_n, T, r(s, d, b))$  então
11:    estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
12:  Senão
13:    Para todo  $m \in M \wedge W_{T_{s,d}} = \infty$  faça
14:       $\kappa(G, C, b_m, P_n)$ 
15:       $(W(T_n), T_n) = \theta(\tilde{G}_{n,b_m}, P_n, r(s, d, b)) \quad \forall n$ 
16:       $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
17:    Fim Para todo
18:    Se  $W_{T_{s,d}} = \infty$  então
19:      bloqueia requisição  $(r(s, d, b))$ 
20:    Senão
21:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
22:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
23:    Fim Se
24:  Fim Se
25: Fim Se

```

linhas 15 e 16 são as responsáveis pela criação do p -cycle com menor custo. Se o p -cycle não puder ser criado, a requisição é bloqueada (linha 19). Caso contrário, a requisição é aceita e o p -cycle, bem como o caminho primário, são estabelecidos (linha 21) e as arestas correspondentes são configuradas para ∞ (linhas 22).

A complexidade do algoritmo APPROVE é analisada da seguinte forma: No pior caso, a modulação escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $M \times (E + V)$. Para encontrar o caminho primário, no pior caso, o algoritmo de Dijkstra é executado M vezes em $C \times (N - b_m)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para o p -cycle, o algoritmo de Suurballe [5] é executado M vezes em $C \times (N - b_m)$ grafos, em que a complexidade de Suurballe é $O(E + V \log V)$. Com isso, o algoritmo APPROVE tem complexidade de $M \times (E + V) + M \times (C \times (N - b_m) \times (E + V \log V)) + M \times (C \times (N - b_m) \times (E + V \log V))$. Como M, N, C e b_m podem ser expressados como constantes, a complexidade do algoritmo é $O(E + V \log V)$.

6.5 O Algoritmo MAROTO

O algoritmo *Minimum interference, Adaptive modulation, Routing, spectrum Overlap and Traffic grooming* (MAROTO) similarmente ao algoritmo APPROVE, também, utiliza *p-cycle*, modulação adaptativa, agregação de tráfego e sobreposição de espectro nos caminhos de backup criados. No entanto, estes algoritmos se distinguem na criação do *p-cycle*, visto que o algoritmo MAROTO prioriza a utilização de *p-cycle* transzonal aos caminho protegido, o que é uma forma de prover interferência mínima na rede.

Algoritmo 14 MAROTO

Entrada: $G, r(s, d, b)$

Saída: Caminho primário e Caminho de backup

```

1: Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
2:    $\Delta(G, C, b_m)$ 
3:    $(W(P_n), P_n) = \alpha(\tilde{G}_{n,b_m}, r(s, d, b)) \quad \forall n \in \sigma$ 
4:    $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
5: Fim Para todo
6:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
7: Se  $W_{P_{s,d}} = \infty$  então
8:   bloqueia requisição  $(r(s, d, b))$ 
9: Senão
10:  Se  $\exists v(P_n, T, r(s, d, b))$  então
11:    estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
12:  Senão
13:    Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
14:       $\kappa(G, C, b_m, P_n)$ 
15:       $(W(T_n), T_n) = \Psi(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
16:       $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
17:    Fim Para todo
18:    Se  $W_{T_{s,d}} = \infty$  então
19:      Para todo  $m \in M \wedge W_{P_{s,d}} = \infty$  faça
20:         $\kappa(G, C, b_m, P_n)$ 
21:         $(W(T_n), T_n) = \theta(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
22:         $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
23:      Fim Para todo
24:      Se  $W_{T_{s,d}} = \infty$  então
25:        bloqueia requisição  $(r(s, d, b))$ 
26:      Fim Se
27:    Fim Se
28:    Se  $W_{T_{s,d}} \neq \infty$  então
29:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
30:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
31:    Fim Se
32:  Fim Se
33: Fim Se

```

No algoritmo MAROTO, as linhas 1 a 6 realizam o mesmo processo de criação do caminho primário que os algoritmos RAMBO e APPROVE. Nele, a linha 1 estabelece todo o conjunto de

modulações que será testado na criação do caminho primário. A linha 2 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação, na possível agregação de tráfego e na sobreposição de espectro. As linhas 3 e 4 computam o caminho com o menor custo a partir dos grafos gerados. O processo de escolha do formato de modulação (linha 1) é repetido até que um caminho seja encontrado ou não exista mais formato de modulação a serem testados. Se o peso do caminho mais curto é ∞ (linha 7), não foi possível encontrar um caminho sob a restrição de contiguidade para a demanda b com a alocação começando com o n -ésimo *slot*. Portanto, a requisição é bloqueada (Linha 8). Caso contrário, um *p-cycle* para proteger o caminho óptico é pesquisado dentre os *p-cycles* ativos que já estão protegendo outras requisições (linha 10). Como no algoritmo APPROVE, para a escolha do *p-cycle*, é levado em consideração que todos os caminhos primários protegidos por um *p-cycle* (P_{T_n}) devem ser disjuntos, os nós finais do caminho primário precisam estar no *p-cycle*, e o possível caminho a ser utilizado pela requisição precisa respeitar a restrição de distancia da modulação utilizada. Além disso, os slots que fazem parte do caminho primário (H_{T_n}) precisam ser disjuntos de todos os slots dos caminhos já protegidos ($H_{P_{T_n}}$) pelo *p-cycle* (T_n). Os caminhos que já estão sendo protegidos pelo *p-cycle* necessitam ser disjuntos do novo caminho primário já selecionado. Além disso, os nós finais do caminho primário precisam estar no *p-cycle* e o possível caminho a ser utilizado pela requisição atender os requisitos mínimos de QoT. Caso exista um *p-cycle*, o caminho óptico é estabelecido (linha 11). Caso contrário, um *p-cycle* com interferência mínima para proteger o caminho óptico deve ser criado. A linha 13 escolhe um formato de modulação pra o *p-cycle* que será criado. A linha 14 transforma o multigrafo em $C \times (N - b_m + 1)$ grafos, baseado no formato de modulação e possíveis agregação de tráfego e sobreposição de espectro. Para evitar gargalos na rede e adotando o conceito de interferência mínima, o algoritmo tenta criar um *p-cycle* transzonal ao caminho a ser protegido (linha 15). Se o *p-cycle* transzonal for criado, o *p-cycle* e o caminho são estabelecidos para satisfazer a requisição (linha 29) e as arestas correspondentes no multigrafo G tenham o seu peso alterado para ∞ (Linha 30 e 31), o que significa que os *slots* foram alocados para o caminho óptico recém-criado. Caso nenhum *p-cycle* transzonal ao caminho possa ser criado, então o algoritmo tenta criar um *p-cycle* menor, que não necessariamente seja transzonal ao caminho primário. As linhas 19 a 22 repetem o processo das linhas 13 a 16, com a diferença do tipo de *p-cycle* criado, que ao invés de considerar as futuras requisições utilizando interferência mínima, considera apenas a requisição atual escolhendo o menor *p-cycle* possível. Caso nenhum *p-cycle* possa ser criado, a requisição é bloqueada (Linha 25). Se o *p-cycle* for criado, o *p-cycle* e o caminho são estabelecidos para satisfazer o pedido (linha 29) e as arestas correspondentes terão o seu peso alterado para ∞ (Linha 30), o que significa que os *slots* foram alocados para o caminho óptico recém-criado.

A complexidade do algoritmo MAROTO é analisada como segue: No pior caso a modulação escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $O(E + V)$. No pior caso, para encontrar o caminho primário o algoritmo de Dijkstra é executado M vezes em $C \times (N - b)$ grafos, onde a complexidade amortizada de Dijkstra é $O(E + V \log V)$. No pior caso, o algoritmo não encontrará um caminho totalmente transzonal ao *p-cycle*, sendo necessário transformar o multigrafo em grafos M vezes, obtendo uma complexidade de $O(E + V)$ e executando o algoritmo de Suurballe [5] M vezes em $C \times (N - b)$ grafos e logo após no pior caso a modulação escolhida será a com menor eficiência espectral, sendo necessário transformar o multigrafo

em grafos M vezes, obtendo uma complexidade de $O(E + V)$ e executando o algoritmo de Suurballe M vezes, sendo a complexidade de Suurballe é $O(E + V \log V)$. Com isso, o algoritmo MAROTO tem complexidade de $M \times (E + V) + M \times (C \times (N - b_m) \times (E + V \log V)) + M \times (C \times (N - b_m) \times (E + V \log V)) + M \times (C \times (N - b_m) \times (E + V \log V))$. Como M , N , C e b_m podem ser expressos como constantes a complexidade do algoritmo é $O(E + V \log V)$.

6.6 Resultados numéricos

Como nos capítulos anteriores, para avaliar o desempenho dos algoritmos propostos neste capítulo, experimentos de simulação foram realizados utilizando o simulador FlexGridSim [59]. Em cada simulação, 100.000 requisições foram geradas, utilizou-se o método de replicação independente e adotou-se nível de confiança de 95% para os intervalos de confiança. Diferentes topologias com diferentes conectividades, foram empregadas nas simulações a Pan-European (Figura 3.2(a)) com 28 nós e 39 enlaces bidirecionais, a USA (Figura 3.2(b)) com 24 nós e 43 enlaces bidirecionais, a NSF (Figura 3.2(c)), com 14 nós e 18 enlaces bidirecionais, as figuras mostram as respectivas distancias entre os nós. O espectro foi dividido em 320 *slots* de espectro e as fibras simuladas possuem 7 núcleos. A carga foi variada entre 25 e 500 erlangs em intervalos de 25 erlangs. Sete tipos de requisições foram utilizadas 25 Gbps, 50 Gbps, 125 Gbps, 200 Gbps, 500 Gbps, 750 Gbps e 1 Tbps. As métricas usadas para avaliar os algoritmos apresentados neste capítulo são as mesmas utilizadas nos capítulos anteriores.

Nas figuras, as curvas rotuladas como RAMBO mostram os resultados para as redes que utilizam o algoritmo RAMBO (Algoritmo 12), as curvas rotuladas como APPROVE mostram os resultados para as redes que utilizam o algoritmo APPROVE (Algoritmo 13) e as curvas rotuladas como MAROTO mostram os resultados para as redes que utilizam o algoritmo MAROTO (Algoritmo 14). Como no capítulo 4, as curvas rotuladas como SSCAM encontram o caminho primário através do algoritmo proposto em [102], utilizando $k = 3$ e o caminho de *backup*, adicionando a capacidade de compartilhamento ao mesmo algoritmo. Além disso, este algoritmo utiliza uma escolha de modulação baseada no tamanho do caminho, enquanto as curvas rotuladas como Cap-DPPM mostram os resultados para as redes que utilizam o algoritmo proposto em [100] e utiliza uma escolha de modulação baseada no tamanho do caminho. Como no capítulo 5, as curvas rotuladas como SBRC mostram os resultados para as redes que utilizam o algoritmo SBRC (Algoritmo 8), as curvas rotuladas como STOP mostram os resultados para as redes que utilizam o algoritmo STOP (Algoritmo 9) e as curvas rotuladas como FRSCA mostram os resultados para as redes que utilizam o algoritmo FRSCA (Algoritmo 10)

A Figura 6.1 mostra a probabilidade de bloqueio para a topologia Pan-European. Os algoritmos SBRC, STOP e FRSCA não utilizam modulação. O algoritmo SBRC inicia o bloqueio de requisições sob carga de 125 erlangs, enquanto os algoritmos SBRC e FRSCA iniciam o bloqueio sob cargas de 150 erlangs. Enquanto os algoritmos Cap-DPPM e o SSCAM iniciam o bloqueio de requisições sob cargas de 100 e 125 erlangs, respectivamente, o algoritmo APPROVE inicia o bloqueio de requisições apenas sob carga de 350 erlangs. Os algoritmos RAMBO e MAROTO por sua vez iniciam o bloqueio de requisições apenas sob cargas de 400 erlangs. Os algoritmos SBRC, STOP, FRSCA, APPROVE, RAMBO e MAROTO utilizam agregação de tráfego e sobreposição de espectro, no entanto, a capacidade de modulação adaptativa provida

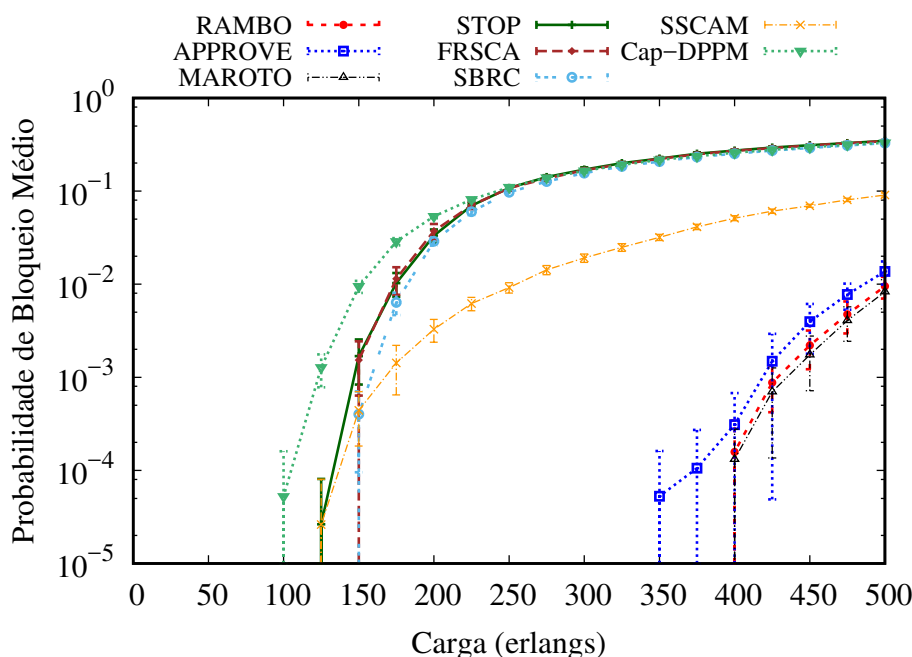


Figura 6.1: Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.

aos algoritmos APPROVE, RAMBO e MAROTO possibilitam a aceitação total de requisições sob cargas altas. Enquanto que o algoritmo SBRC, que utiliza compartilhamento de caminho de backup, produz valores de BBR menores que os algoritmos STOP e FRSCA, que utilizam p -cycle FIPP e p -cycle FIPP e interferência mínima, respectivamente, com a adição de modulação adaptativa o algoritmo MAROTO que utiliza p -cycle FIPP e interferência mínima produz valores menores de BBR que os algoritmos RAMBO e APPROVE, respectivamente. Os menores bloqueios gerados pelos algoritmos MAROTO, RAMBO e APPROVE com relação aos algoritmos Cap-DPPM e SSCAM que também utilizam modulação adaptativa estão relacionados ao fato destes algoritmos utilizarem agregação de tráfego e sobreposição de espectro nos caminhos de *backup*. Além disso estes algoritmos utilizam a técnica do multigrafo apresentada no Capítulo 3 para encontrar seu caminhos, o que leva à uma melhor escolha dos caminhos. A utilização de sobreposição de espectro e modulação adaptativa para caminhos de *backup*, evita o desperdício na utilização de recursos de proteção, o que implica em uma maior disponibilidade de recursos para novas requisições geradas. Os valores de BBR produzidos pelo algoritmo MAROTO é sempre menor que o dos outros algoritmos avaliados para todas as cargas simuladas, o que evidencia os benefícios de considerar modulação adaptativa, agregação de tráfego e sobreposição de espectro juntamente com interferência mínima para criar p -cycles ao escolher a rota de backup em topologias com alta conectividade. Apesar do algoritmo APPROVE produzir valores de BBR maiores do que os valores produzidos pelo algoritmo RAMBO, a utilização do algoritmo APPROVE é interessante considerando seu menor tempo de comutação de proteção. A alta conectividade dos nós da topologia Pan-European permitiu que o algoritmo MAROTO, que utiliza uma técnica de interferência mínima, produza menor bloqueio do que os algoritmos APPROVE e RAMBO, pois ao utilizar interferência mínima a criação de caminhos de backup evitam o gargalo de enlaces na rede.

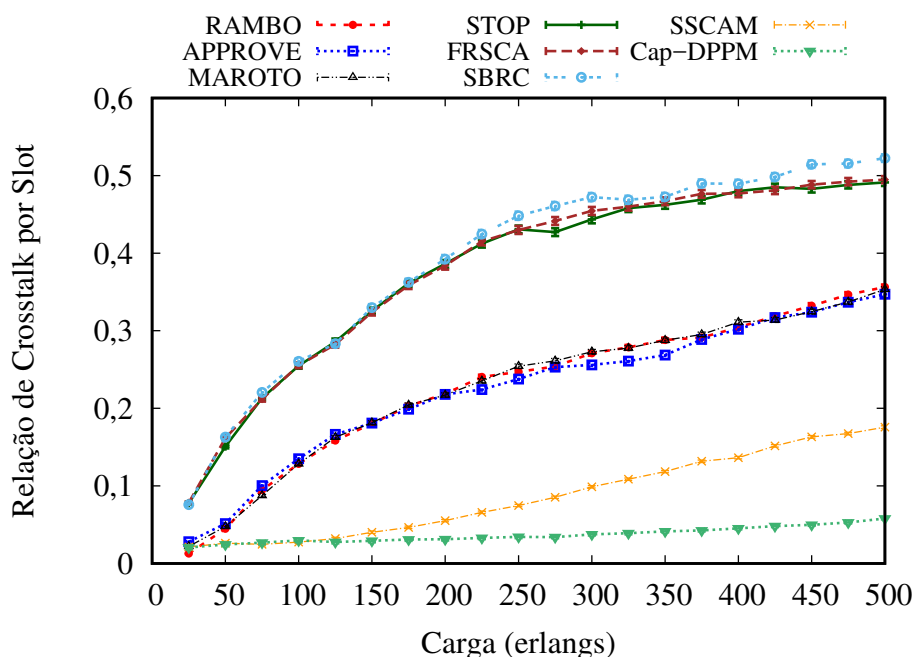


Figura 6.2: *Crosstalk* por *slot* dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.

A Figura 6.2 mostra a relação de *crosstalk* por *slot* gerado para a topologia Pan-European. Os valores de CpS gerados pelos algoritmos MAROTO, RAMBO e APPROVE iniciam com um valor de 0,02 e aumentam até 0,34. O algoritmo Cap-DPPM produz valores menores de CpS iniciando em 0,02 e aumentando até 0,05. Os algoritmos SBRC, STOP e FRSCA produzem os maiores valores de CpS iniciando em 0,08 e aumentando até 0,5. O alto CpS produzidos por estes algoritmos ocorre devido a utilização de modulação fixa, com isso um maior número de slots sofre com *crosstalk*, devido ao menor número de bits por símbolos aceito em cada conexão. Os algoritmos MAROTO, RAMBO e APPROVE produzem valores mais altos de CpS do que os algoritmos Cap-DPPM e SSCAM. Isto ocorre pois estes algoritmos aceitam mais conexões do que os outros algoritmos, sendo estas conexões responsáveis pelo maior *crosstalk* gerado. Apesar da utilização de modulação adaptativa o CpS gerado pelos algoritmos MAROTO, RAMBO e APPROVE aumentam rapidamente com o aumento da carga, uma vez que o número de saltos usados leva a uma maior utilização de recursos e consequentemente mais CpS. Apesar dos algoritmos APPROVE e MAROTO que utilizam *p-cycle* utilizarem mais recursos nos caminhos de *backup*, o que deveria dificultar a criação de novos caminhos primários, o alto compartilhamento desses caminhos e a sobreposição de espectro utilizada permitiram que os valores de *crosstalk* gerados fossem próximos ao gerados pelo algoritmo RAMBO.

A Figura 6.3 mostra a razão de fragmentação gerada para a topologia Pan-European. Para todas as cargas simuladas, o algoritmo RAMBO produz a menor razão de fragmentação, isto acontece pois este algoritmo possui uma menor e melhor alocação dos recursos, motivado pela utilização do multigrafo que encontra caminhos disponíveis menores, da modulação adaptativa que permite que caminhos menores utilizem um maior número de bits por símbolo, e da sobreposição de espectro que permite que *slots* possa ser sobrepostos. Sob altas cargas, os algoritmos SBRC, STOP e FRSCA que não utilizam modulação adaptativa produzem a maior relação

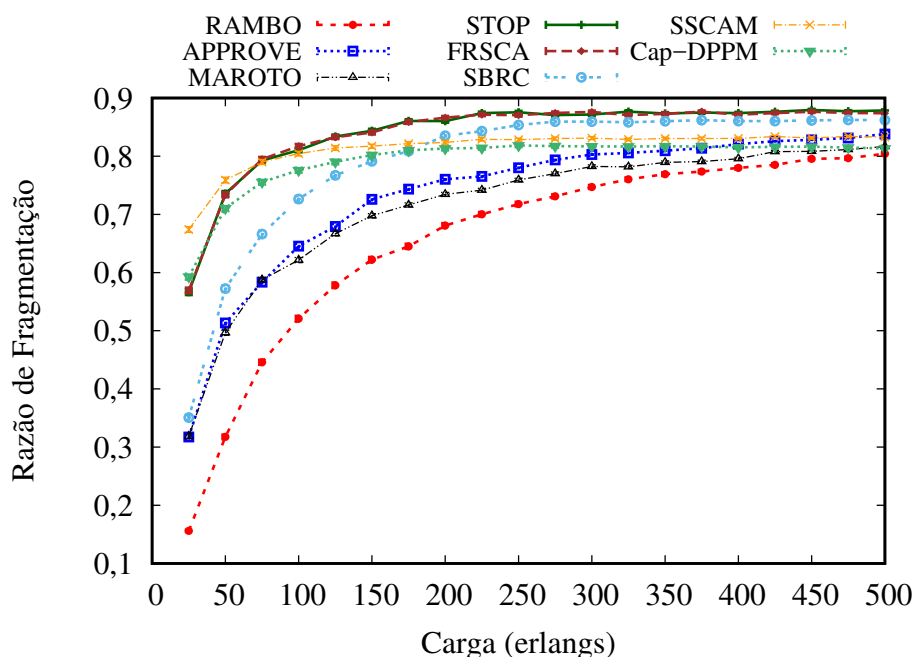


Figura 6.3: Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.

de fragmentação, devido a maior necessidade de slots necessários consequentemente aumentando o número de faixas de espectros disponíveis que não podem ser utilizadas. Os algoritmos RAMBO, APPROVE e MAROTO produzem grande impacto nos valores de CpS quando comparado com os algoritmos SBRC, APPROVE e MAROTO que não utilizam modulação. O algoritmo MAROTO possui razão de fragmentação menor que o algoritmo APPROVE, devido ao maior compartilhamento gerado pelo algoritmo MAROTO diminuindo o estabelecimento de caminhos de *backup* e por sua vez justificando a menor BBR gerada. O algoritmo Cap-DPPM produz a menor razão de fragmentação, uma vez que ele aceita um menor número de requisições, gerando menos fragmentação. A utilização de sobreposição de espectro juntamente com modulação adaptativa pelos algoritmos RAMBO, APPROVE e MAROTO permite que seja realizado uma melhor alocação dos recursos e consequentemente gerar menor fragmentação que os outros algoritmos que não utilizam sobreposição de espectro e modulação adaptativa de forma conjunta.

A Figura 6.4 mostra a eficiência energética gerada para a topologia Pan-European. O algoritmo Cap-DDPM produz a maior eficiência energética dos algoritmos avaliados, isto ocorre pois este algoritmo produz o maior bloqueio, gerando menor uso dos recursos e consequentemente uma maior eficiência energética. Os algoritmos SBRC, STOP e FRSCA produziram a menor eficiência energética, pois a modulação adaptativa utiliza menor número de recursos da rede. Apesar do algoritmo SSCAM produzir alto bloqueio, ele também produz a menor eficiência energética quando comparado aos algoritmos que utilizam modulação adaptativa, isto ocorre pois este algoritmo possui o maior o número médio de saltos do caminho primário principalmente sob cargas altas. O algoritmo RAMBO produz maior eficiência energética que os algoritmos APPROVE e MAROTO que utilizam p-cycle FIPP.

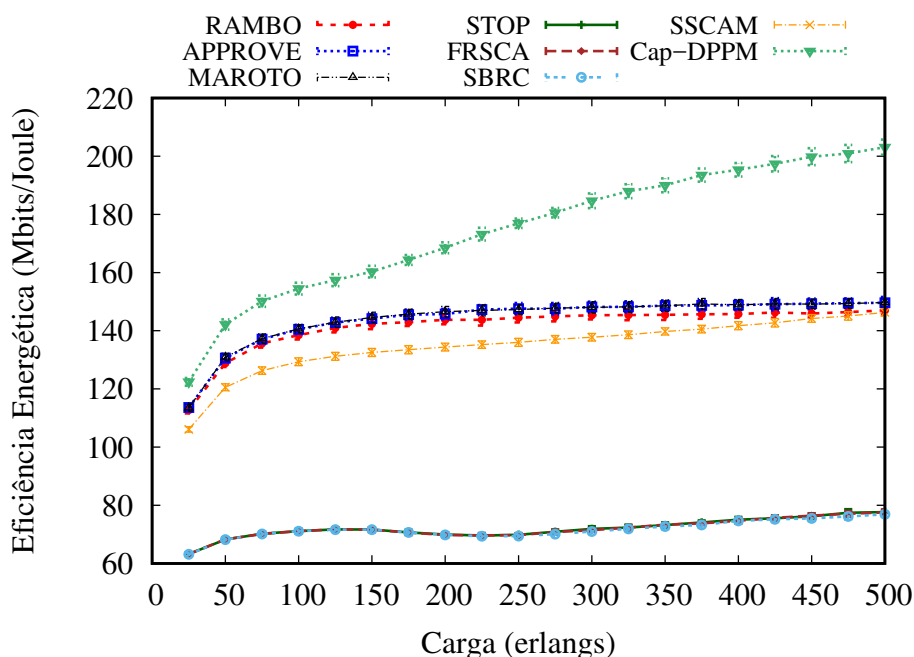


Figura 6.4: Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.

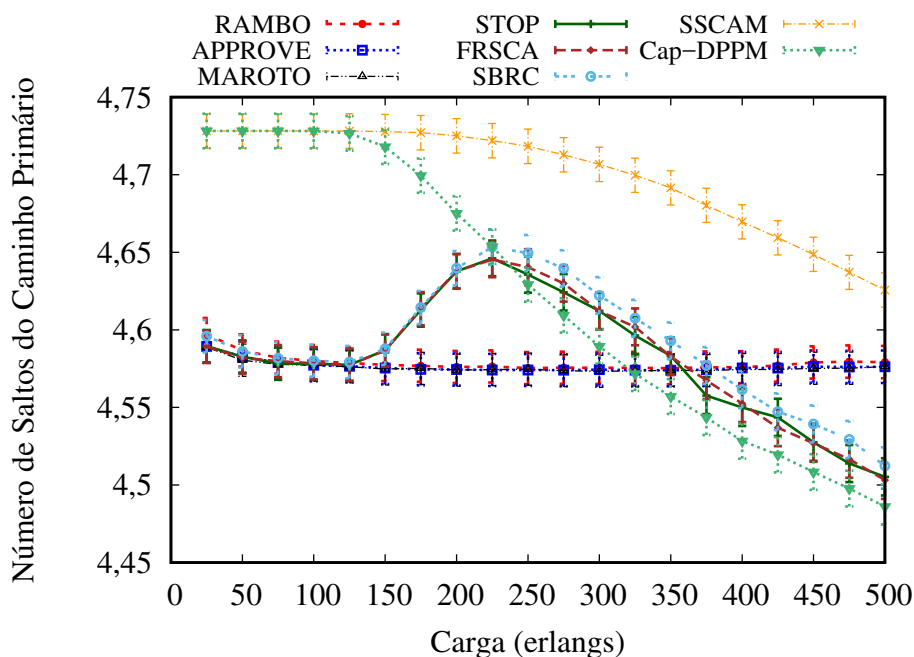


Figura 6.5: Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.

A Figura 6.5 mostra o número médio de saltos do caminho primário gerado para a topologia Pan-European. Sob cargas baixas, os algoritmos SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO possuem menor número médio de saltos do caminho primário, isto ocorre pois estes algoritmos utilizam a técnica do multigrafo e agregação de tráfego. Com o aumento da carga, os algoritmos RAMBO, APPROVE e MAROTO que utilizam modulação adaptativa conseguem

manter um baixo bloqueio de requisições e ainda um número médio de saltos do caminho primário com pouca variação de tamanho. Por outro lado, sob altas cargas, os algoritmos SBRC, STOP e FRSCA aumentam o número de requisições bloqueadas e conseqüentemente decrementam o número de saltos de caminhos primários. Apesar de produzir altos valores BBR, os algoritmos Cap-DPPM e SSCAM, que utilizam modulação adaptativa, produzem um alto número médio de saltos do caminho primário em comparação com os outros algoritmos simulados. Isto acontece pois os algoritmos Cap-DPPM e SSCAM não utilizam sobreposição de espectro, agregação de tráfego e a técnica do multigrafo. Conforme aumenta a carga da rede os algoritmos SSCAM e Cap-DPPM diminuem o número médio de saltos do caminho primário. Isto ocorre devido ao aumento do bloqueio de conexões e, conseqüentemente, apenas caminhos menores são estabelecidos.

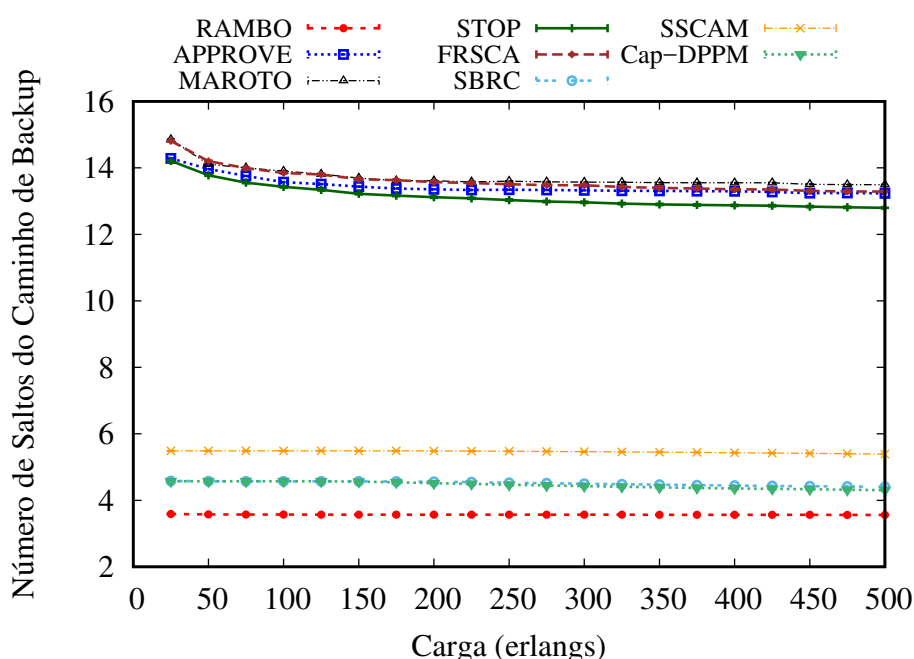


Figura 6.6: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia Pan-European.

A Figura 6.6 mostra o número médio de saltos do caminho de *backup* gerado para a topologia Pan-European. Como esperado os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com maior número médio de saltos do que os caminhos que não utilizam. Os algoritmos FRSCA e MAROTO, que utilizam *p-cycle* FIPP e interferência mínima, produzem a maior média de número de backup, seguidos pelos algoritmos STOP e APPROVE, que também utilizam *p-cycle* FIPP. O algoritmo SBRC, que não utiliza modulação adaptativa, produz caminhos de *backup* com número médio de saltos próximo aqueles dos algoritmos que utilizam modulação adaptativa. O algoritmo MAROTO produz caminhos de *backup* com número médio de saltos sempre maior que o algoritmo APPROVE, pois o algoritmo MAROTO prioriza a criação de *p-cycles* transzonais ao caminho primário sempre que possível, com a finalidade de evitar gargalos nos enlaces da rede. Os *p-cycle* transzonais aos caminhos primário tendem a ser maiores que os caminhos que usam os mesmos enlaces do *p-cycle*. O algoritmo RAMBO

produz caminhos de *backup* com o menor número médio de saltos. Os algoritmos SSCAM e CaP-DPPM produzem maior número médio de saltos do caminho de *backup* do que o algoritmo RAMBO a não utilização de agregação de tráfego e sobreposição de espectro.

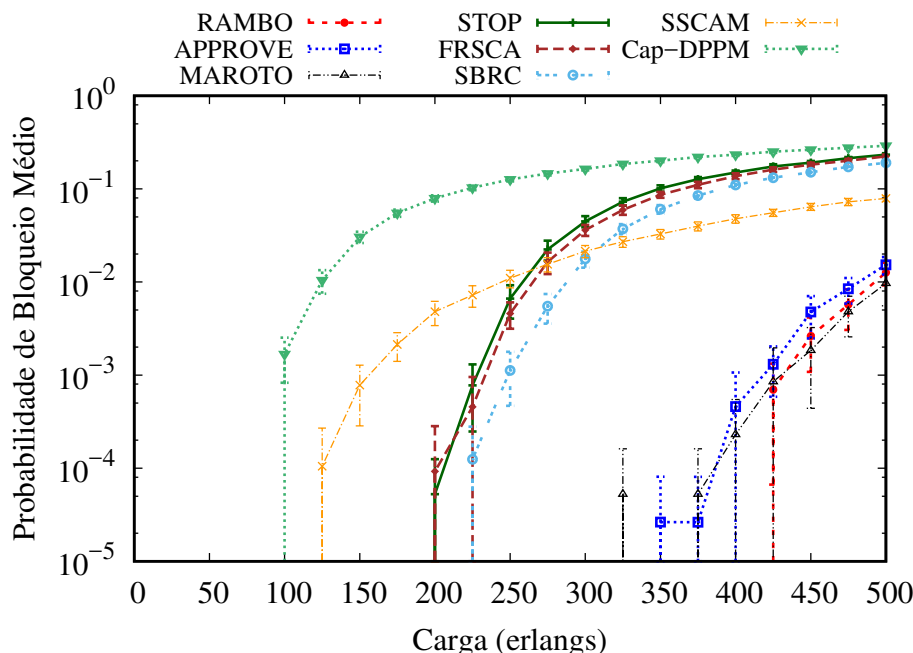


Figura 6.7: Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.

A Figura 6.7 mostra a probabilidade de bloqueio para a topologia USA. Enquanto o Cap-DPPM e o SSCAM iniciam o bloqueio de requisições sob cargas de 100 e 125 erlangs, respectivamente, os algoritmos MAROTO e APPROVE iniciam o bloqueio de requisições apenas sob cargas de 325 e 350 erlangs, respectivamente. Por sua vez, o algoritmo RAMBO inicia o bloqueio somente sob cargas de 425 erlangs. Enquanto os algoritmos FRSCA e STOP iniciam o bloqueio de requisições sob cargas de 200 erlangs, o algoritmo SBRC inicia o bloqueio de requisições sob 225 erlangs. O BBR produzido pelos algoritmos RAMBO, APPROVE e MAROTO, que utilizam modulação adaptativa, é bem menor quando comparados com os algoritmos SBRC, STOP e FRSCA, o que demonstra o impacto de combinar as técnicas de modulação adaptativa, agregação de tráfego e sobreposição de espectro em um algoritmo. Diferentemente da topologia Pan-European, nesta topologia, o algoritmo RAMBO produz os menores valores de BBR. O baixo valor de BBR produzido pelo algoritmo RAMBO evidencia a vantagem de se utilizar caminhos de backup compartilhados, modulação adaptativa, agregação de tráfego e sobreposição de espectro em uma topologia em que com exceção de alguns nós de bordas, possuem pelo menos outros três nós conectados a eles. O algoritmo Cap-DPPM produz o maior valor de BBR devido ao algoritmo CaP-DPPM realizar proteção 1:1, o que gera reserva de recursos individuais para todos os caminhos estabelecidos, e, conseqüentemente impossibilita o estabelecimento de algumas conexões. Além disso, a utilização do algoritmo de k caminhos, com $k=2$, para escolha do caminho primário e de backup, inviabiliza a formação de alguns caminhos gerando bloqueio mesmo com recursos disponíveis. A BBR gerada pelo algoritmo SSCAM, por sua vez, diferencia da BBR gerada pelos algoritmos RAMBO, APPROVE e MAROTO pela

utilização do multigrafo que realiza eficientemente o estabelecimento de recursos e pela utilização da agregação de tráfego e sobreposição de espectro. Além disso, o algoritmo APPROVE inicia o bloqueio sob cargas menores que o algoritmo MAROTO, devido a dificuldade de criar caminhos em determinados nós da rede.

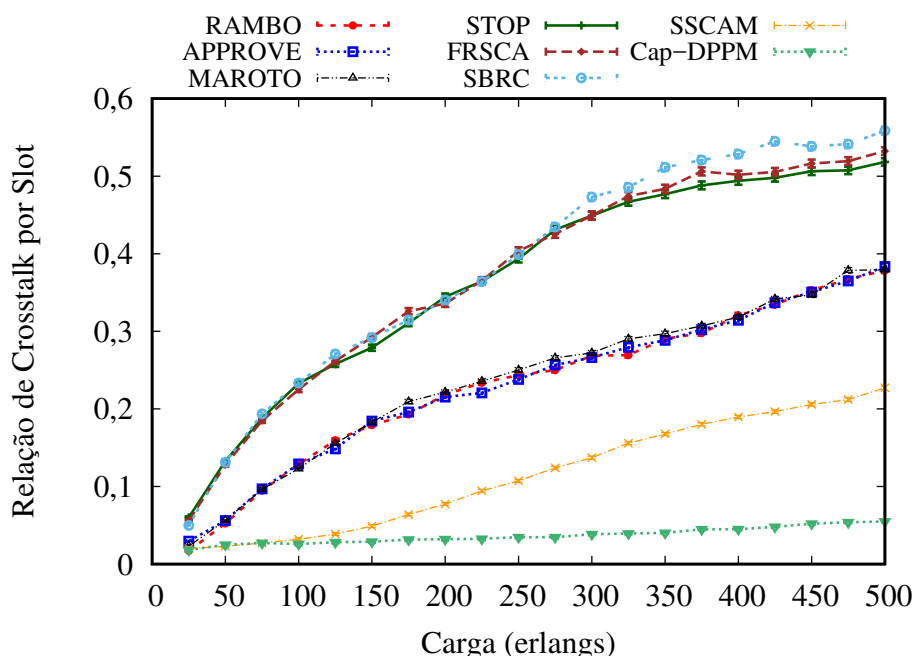


Figura 6.8: *Crosstalk* por *slot* dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.

A Figura 6.8 mostra a relação de *crosstalk* por *slot* gerado para a topologia USA. Os valores de CpS gerados pelos os algoritmos RAMBO, APPROVE e MAROTO iniciam em 0,02 e aumentam até 0,38. Os valores de CpS gerados pelo algoritmo SSCAM iniciam em de 0,02 e aumentam até 0,22. Os valores de CpS gerados pelo algoritmo Cap-DPPM iniciam em 0,01 e permanecem quase constante aumentando apenas até o valor de 0,05. Os valores de CpS gerados pelos algoritmos STOP, FRSCA e SBRC iniciam em 0,06 e aumentam até 0,51, 0,52 e 0,56, respectivamente. Os algoritmos STOP, FRSCA e SBRC produzem o maior valor de CpS dos algoritmos avaliados. Isto ocorre pois estes algoritmos utilizam modulação fixa, utilizando um maior número de *slots* nas conexões. Os algoritmos RAMBO, APPROVE e MAROTO produzem maiores valores de CpS que os algoritmos SSCAM e Cap-DDPM que utilizam modulação adaptativa. No entanto, o *crosstalk* entre núcleos desses algoritmos encontram-se em limiar aceitável. O maior CpS gerado por esses algoritmos deve-se ao número de requisições aceitas, influenciadas pelas técnicas utilizadas, tornando possível uma maior utilização dos recursos disponíveis na rede. O baixo CpS apresentado pelos algoritmos CaP-DPPM e SSCAM é consequência do alto bloqueio gerado por eles. Conforme um menor número de requisições são aceitas um menor número de slots sofre *crosstalk*.

A Figura 6.9 mostra a razão de fragmentação gerada para a topologia USA. O algoritmo RAMBO produz a menor razão de fragmentação como consequência da melhor distribuição de caminhos na alocação de recursos na rede e consequentemente produzindo menor bloqueio quando comparado com os outros algoritmos avaliados. O algoritmo MAROTO possui razão

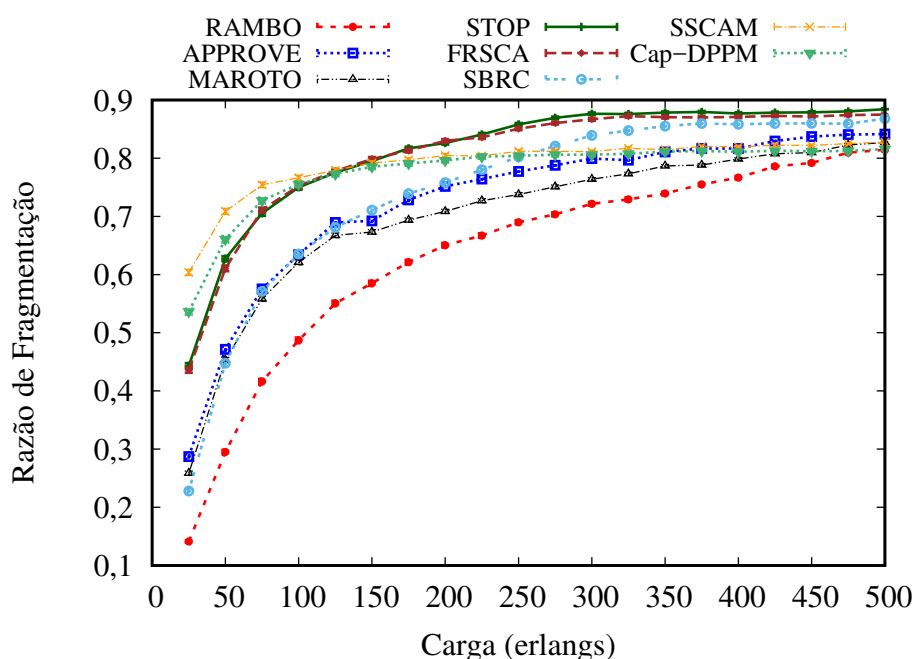


Figura 6.9: Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.

de fragmentação menor que o algoritmo APPROVE, devido ao maior compartilhamento gerado pelo algoritmo MAROTO diminuindo o estabelecimento de caminhos de backup. Diferentemente da topologia Pan-European em que os p-cycles gerados pelo algoritmo MAROTO são mais compartilhados, compensando o maior quantidade recursos requeridos, na topologia USA esse compartilhamento não é o suficiente, gerando BBR sob cargas menores. Os algoritmos RAMBO, APPROVE e MAROTO, que utilizam modulação adaptativa produzem maior razão de fragmentação do que os algoritmos SBRC, STOP e FRSCA, respectivamente, que não utilizam modulação adaptativa. O algoritmo SBRC que utiliza modulação fixa produz razão de fragmentação próxima aos algoritmos APPROVE e MAROTO, que utilizam modulação adaptativa. Isto acontece devido ao maior número de saltos utilizado pelos p-cycles FIPP. O algoritmo Cap-DPPM produz razão de fragmentação próxima à dada pelo algoritmo SSCAM como consequência da não utilização das técnicas utilizadas pelos algoritmos RAMBO, APPROVE e MAROTO.

A Figura 6.10 mostra a eficiência energética gerada para a topologia USA. Os algoritmos RAMBO, APPROVE e MAROTO que utilizam multigrafo, modulação adaptativa, agregação de tráfego e sobreposição de espectro possuem eficiência energética semelhante. O mesmo acontece com os algoritmos SBRC, STOP e FRSCA que utilizam multigrafo, agregação de tráfego e sobreposição de espectro. Os algoritmos RAMBO, APPROVE e MAROTO produzem maior eficiência energética do que os algoritmos SBRC, STOP e FRSCA que não utilizam modulação adaptativa. Para todas as cargas simuladas o algoritmo Cap-DPPM produz a maior eficiência energética como consequência do alto bloqueio de requisições, permitindo que apenas caminhos menores sejam estabelecidos o que implica em um menor consumo energético. Sob cargas menores que 350 erlangs, apesar do algoritmo SSCAM produzir bloqueio menor que os dos algoritmos RAMBO, APPROVE e MAROTO, o algoritmo SSCAM produz cami-

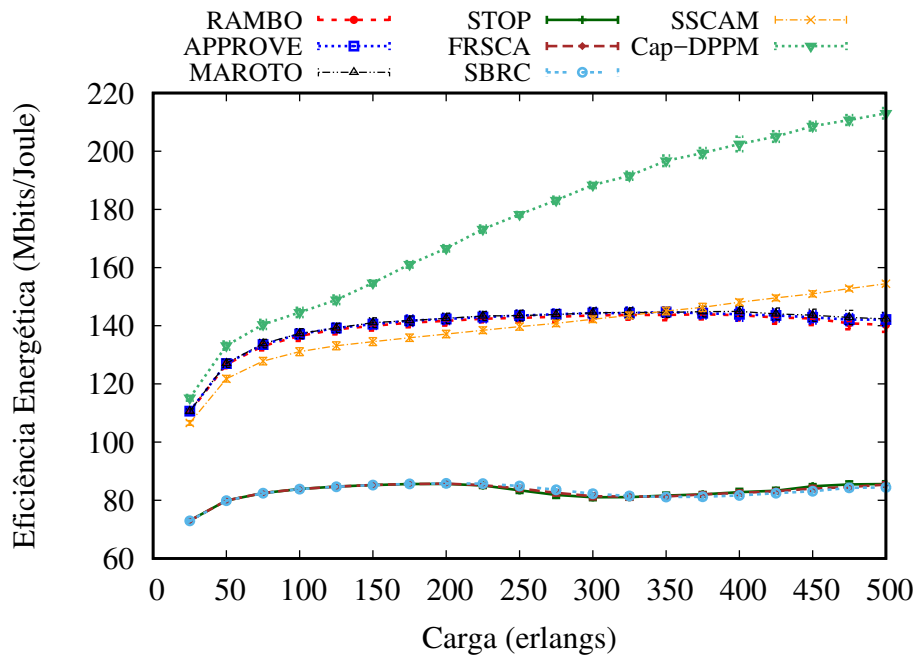


Figura 6.10: Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.

nhos primários maiores influenciando diretamente na eficiência energética gerada. Para cargas maiores que 350, os algoritmos RAMBO, APPROVE e MAROTO aumentam os respectivos número médio de saltos do caminho primário, para manter o baixo bloqueio e a alta utilização da rede, diminuindo eficiência energética.

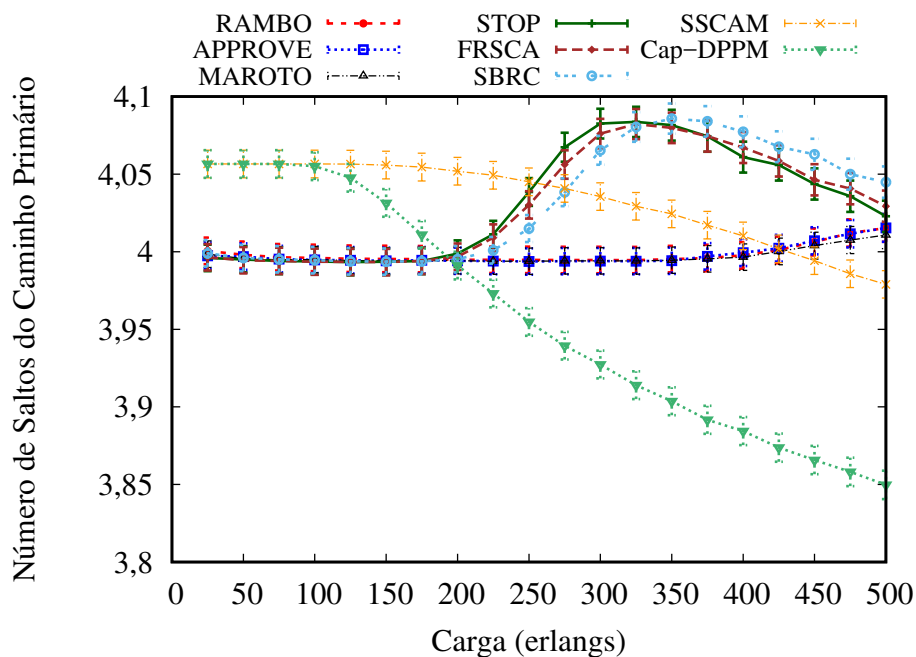


Figura 6.11: Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.

A Figura 6.11 mostra o número médio de saltos do caminho primário gerado para a topologia USA. Os algoritmos RAMBO, APPROVE e MAROTO possuem número médio de saltos do caminho primário semelhantes para qualquer carga simulada, isto ocorre pois eles utilizam a mesma técnica para criação dos caminhos primários e para criação dos caminhos de *backup*. Além disso, utiliza agregação de tráfego, sobreposição de espectro e modulação adaptativa. Os algoritmos SBRC, STOP e FRSCA produzem caminhos primários semelhantes para qualquer carga simulada, pois eles utilizam as mesmas técnicas na criação do caminho. Sob cargas baixas, os algoritmos RAMBO, APPROVE, MAROTO, SBRC, STOP e FRSCA produzem caminhos primários semelhantes conforme aumenta-se a carga da rede os algoritmos SBRC, STOP e FRSCA aumentam o número de saltos dos caminhos, devido a não utilização de modulação adaptativa. A alta eficiência destes algoritmos permite que o número de saltos mantenha-se quase constante, produzindo incremento apenas sob altas cargas. O número médio de saltos do algoritmo Cap-DPPM decreta conforme ocorre o aumento da carga na rede. Por sua vez, sob cargas menores que 425 erlangs, o algoritmo SSCAM produz o maior número médio de saltos do caminho primário, influenciado pelo principalmente pela forma de criação dos caminhos.

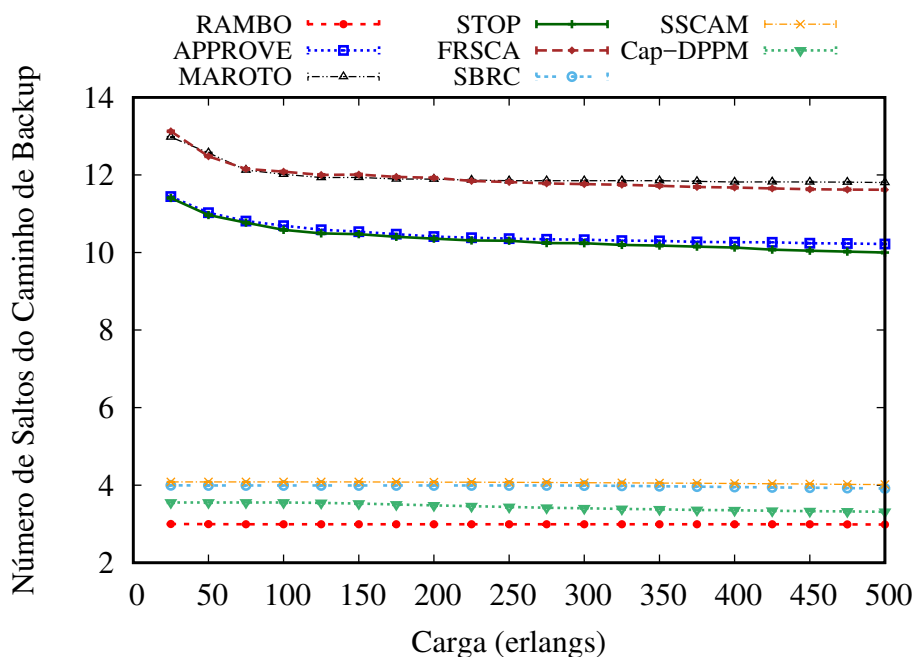


Figura 6.12: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia USA.

A Figura 6.12 mostra o número médio de saltos do caminho de *backup* gerado para a topologia USA. Como para as outras topologias, os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com mais que três vezes o número médio de saltos que os caminhos que não o utilizam. O algoritmo APPROVE, que utiliza modulação adaptativa, produz número de saltos dos caminhos de *backup* semelhantes aqueles produzidos pelo algoritmo STOP, que utiliza modulação fixa. Por sua vez, o algoritmo MAROTO, que também utiliza modulação adaptativa, produz número de saltos dos caminhos de *backup* semelhantes aqueles produzidos

pelo algoritmo FRSCA, que utiliza modulação fixa. Por outro lado, o algoritmo RAMBO, que utiliza modulação adaptativa, produz número de saltos dos caminhos de *backup* sempre menor que aqueles produzidos pelo algoritmo SBRC que utiliza modulação fixa. O algoritmo RAMBO possui o menor número médio de saltos do caminho de backup. Isto ocorre pois este algoritmo além de realizar uma melhor escolha do caminho através da técnica do multigrafo, possui uma maior disponibilidade dos recursos como consequência das técnicas de agregação de tráfego e sobreposição de espectro. Considerando os algoritmos que utilizam *p-cycle*, o algoritmo APPROVE é o que aloca o menor número médio de saltos do caminho de *backup*, isto ocorre pois este algoritmo procura utilizar o menor *p-cycle* para realizar a proteção do caminho ao contrário do MAROTO que visa evitar congestionamento dos enlaces. Os algoritmos Cap-DPPM e SSCAM produzem maior número médio de saltos para o caminho de *backup* que o algoritmo RAMBO, isto ocorre pois o algoritmo RAMBO possui maior disponibilidade dos recursos como consequência do uso de agregação de tráfego e sobreposição de espectro.

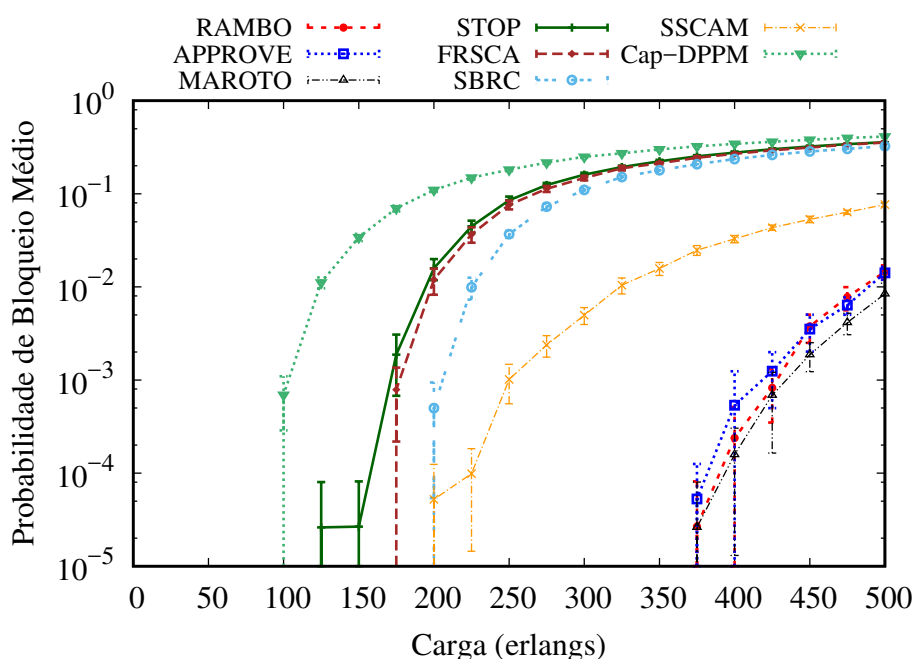


Figura 6.13: Bloqueio de banda dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.

A Figura 6.13 mostra a probabilidade de bloqueio para a topologia NSF. Mesmo com o baixo grau dos nós nesta topologia a utilização de modulação adaptativa, agregação de tráfego e sobreposição de espectro evita a criação de gargalos e o aumento muito mais rápido nas BBRs produzidas pelos algoritmos RAMBO, APPROVE e MAROTO quando comparado com os outros algoritmos avaliados. Enquanto os algoritmos STOP, FRSCA e SBRC iniciam o bloqueio de requisições, respectivamente, sob carga de 125, 175 e 200 erlangs, os algoritmos RAMBO, APPROVE e MAROTO iniciam o bloqueio de requisições apenas sob cargas de 375 erlangs. Isto evidencia as vantagens de combinar modulação adaptativa, agregação de tráfego e sobreposição de espectro, principalmente, quando utilizado juntamente com *p-cycle* FIPP em topologias com baixa conectividade. Os algoritmos Cap-DPP e SSCAM iniciam o bloqueio de requisições respectivamente, sob carga de 100 e 200 erlangs. Apesar do algoritmo Cap-DPPM

utilizar modulação adaptativa, produz valores maiores de BBR do que os algoritmos STOP, SBRC e FRSCA, que utilizam agregação de tráfego e sobreposição de espectro, mas utilizam modulação fixa. Em decorrência do uso das técnicas de modulação adaptativa, agregação de tráfego e sobreposição de espectro o algoritmo MAROTO produz menor BBR mesmo com a geração de p -cycle sendo mais difícil para as topologias com menor conectividade. Entre cargas de 375 e 450 erlangs, o algoritmo RAMBO produz menor BBR que o algoritmo APPROVE, como consequência do maior uso dos recursos gerado pela criação de p -cycles. Entre 450 e 500 erlangs o algoritmo MAROTO produz um maior compartilhamento, consequentemente, gerando menores valores de BBR que o algoritmo RAMBO. Os algoritmos Cap-DPPM produzem o bloqueio devido à forma de criação dos caminhos. O algoritmo SSCAM produz o segundo maior bloqueio, menor que o algoritmo Cap-DPPM devido ao compartilhamento do caminho de *backup* e menor que os outros algoritmos tanto por não utilizar o multigrafo quanto por não utilizar agregação de tráfego e sobreposição de espectro.

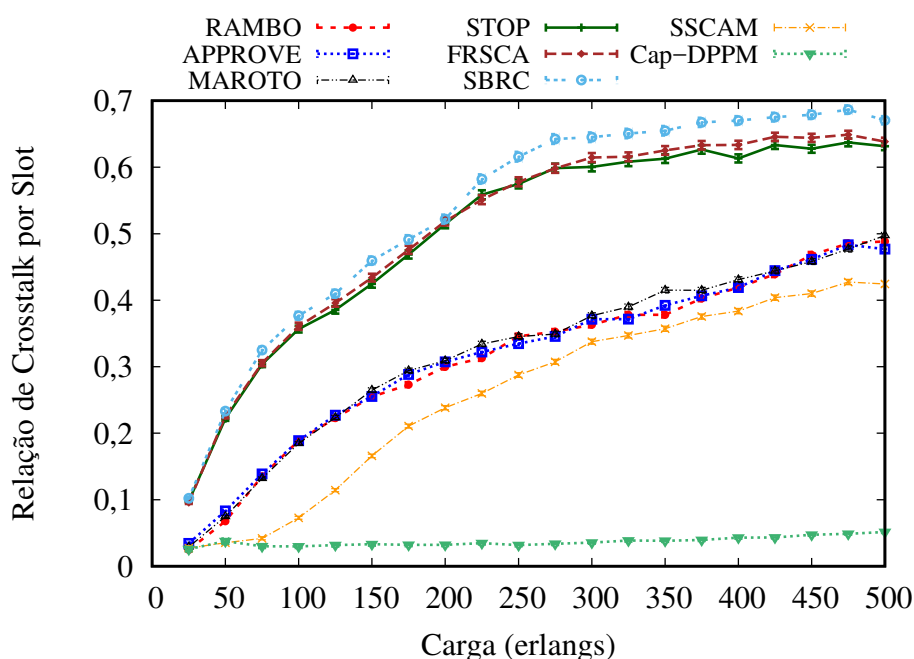


Figura 6.14: *Crosstalk* por *slot* dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.

A Figura 6.14 mostra a relação de *crosstalk* por *slot* gerado para a topologia NSF. Diferente das outras topologias simuladas os valores de CpS gerado pelo algoritmo SSCAM é bem próximo aos gerados pelos algoritmos RAMBO, APPROVE e MAROTO, apesar do alto bloqueio gerado pelo algoritmo SSCAM. O CpS dos algoritmos RAMBO, APPROVE e MAROTO iniciam em 0,03 e aumentam até 0,47. O CpS do algoritmo SSCAM inicia com um valor de 0,02 e aumenta até 0,17. O algoritmo Cap-DPPM produz os valores de CpS mais baixos quando comparados aos valores de CpS gerados pelos outros algoritmos, como consequência do alto BBR produzido e da baixa utilização. Os algoritmos STOP, FRSCA e SBRC produzem os valores de CpS mais altos quando comparados aos valores de CpS gerados pelos outros algoritmos. O CpS desses algoritmos iniciam com um valor de 0,1 e aumenta até 0,64 para os algoritmos FRSCA e STOP e até 0,68 para o algoritmo SBRC. Os valores de CpS dos algoritmos STOP,

FRSCA e SBRC são sempre mais altos do que aqueles produzidos pelos algoritmos RAMBO, APPROVE e MAROTO. Isto acontece pela utilização de modulação fixa feita pelos algoritmos STOP, FRSCA e SBRC.

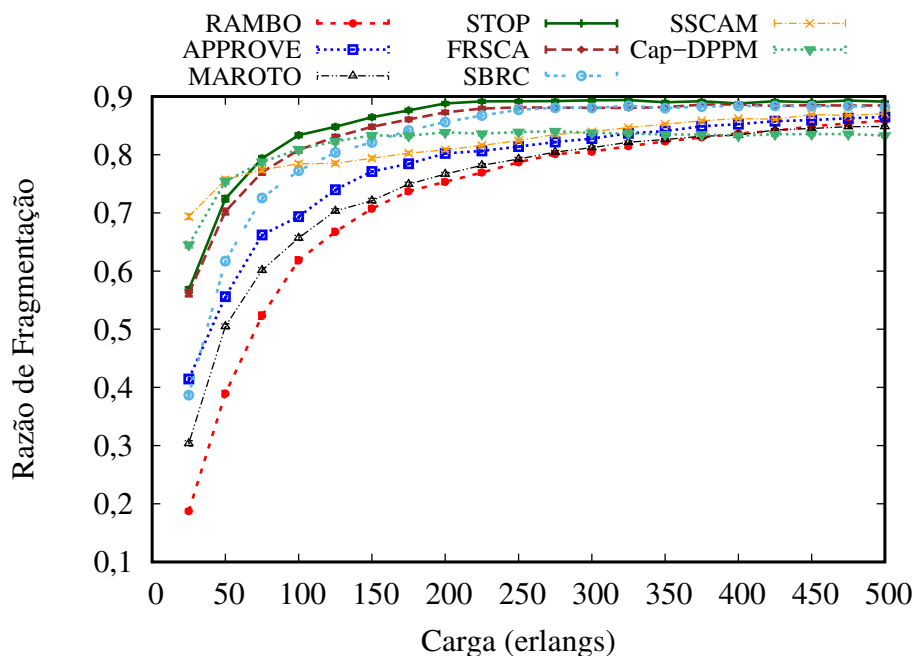


Figura 6.15: Razão de fragmentação dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.

A Figura 6.15 mostra a razão de fragmentação gerada para a topologia NSF. Para cargas menores que 350 erlangs, o algoritmo RAMBO produz a menor razão de fragmentação, consequentemente, não produz valores de BBR sob essas cargas. Sob altas cargas, os algoritmos STOP, FRSCA e SBRC produzem os maiores valores de fragmentação, o que justifica o alto bloqueio produzido por estes algoritmos. Isto acontece por eles não utilizarem modulação adaptativa, consequentemente, necessitam de mais recursos por conexão. A baixa razão de fragmentação gerada pelos algoritmos RAMBO, APPROVE e MAROTO é consequência da utilização da modulação adaptativa, da agregação de tráfego e sobreposição de espectro em conjunto, tornando os algoritmos mais eficientes que os que não utilizam.

A Figura 6.16 mostra a eficiência energética gerada para a topologia NSF. A eficiência energética produzida pelo algoritmo CaP-DPPM é maior que a produzida pelos outros algoritmos, pois este algoritmo produz um alto bloqueio gerado pela forma de escolha dos caminhos. Os algoritmos STOP, FRSCA e SBRC produzem os menores valores de eficiência energética, pois estes algoritmos não utilizam modulação adaptativa, consequentemente, utilizando um maior número de recursos que influencia no consumo energético. Não há muita diferença entre a eficiência energética produzida pelos algoritmos APPROVE e MAROTO, isto reflete o número médio de saltos do caminho primário que eles possuem. Como para os algoritmos RAMBO, APPROVE e MAROTO, não há muita diferença entre a eficiência energética produzida pelos algoritmos STOP, FRSCA e SBRC, isto reflete o número médio de saltos do caminho primário que eles possuem. O algoritmo SSCAM possui a menor eficiência energética, desde que este algoritmo possui o maior número médio de saltos do caminho primário.

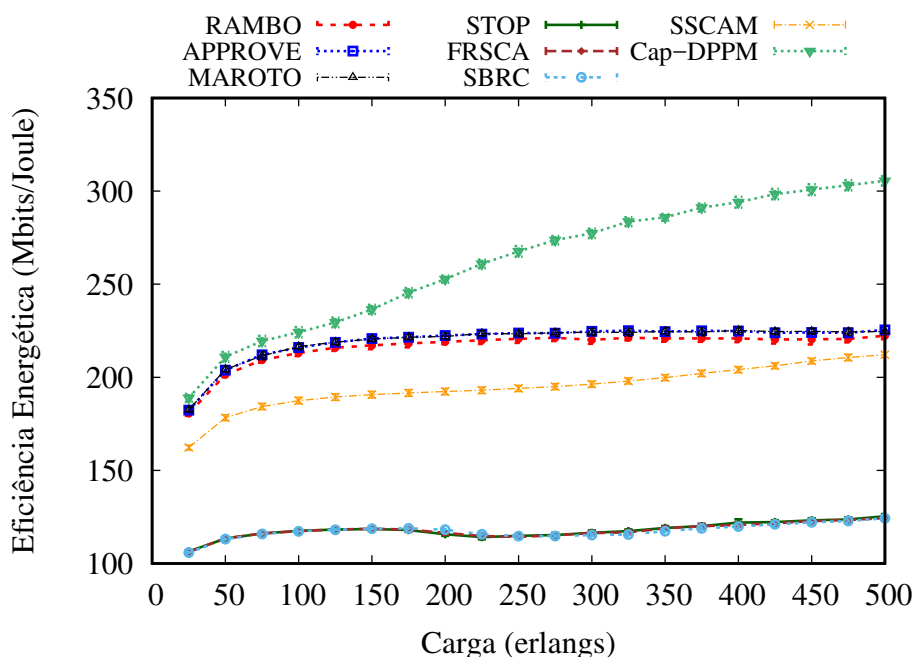


Figura 6.16: Eficiência Energética dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.

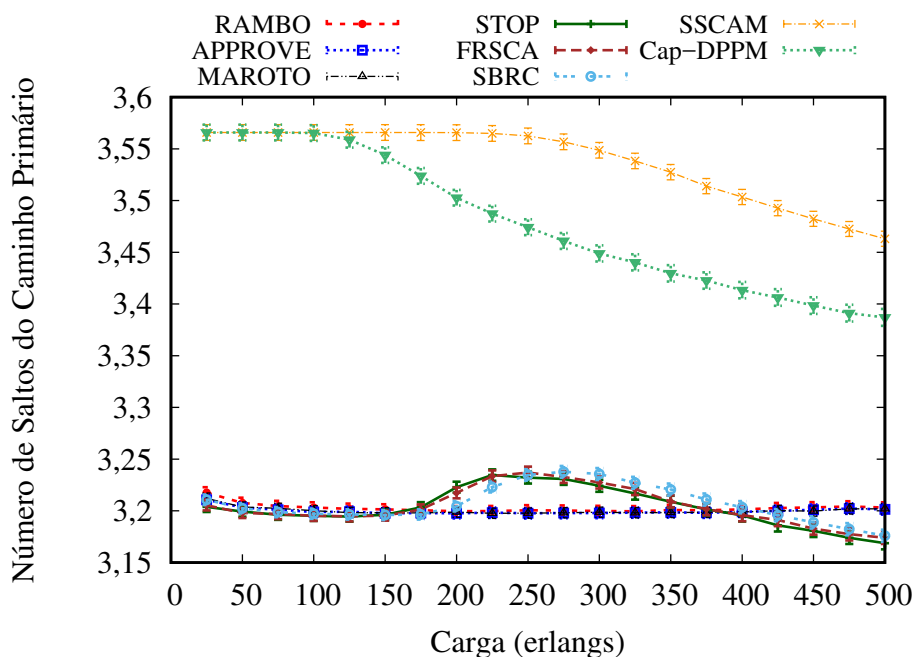


Figura 6.17: Número médio de saltos do caminho primário dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.

A Figura 6.17 mostra o número médio de saltos do caminho primário gerado para a topologia NSF. Os algoritmos RAMBO, APPROVE e MAROTO que utilizam o multigrafo possuem sempre menor número médio de saltos que os caminhos que não utilizam o multigrafo. O algoritmo SSCAM produz caminhos com o maior número médio de saltos quando comparados com outros algoritmos. Como nas topologias anteriores, os algoritmos RAMBO, APPROVE

e MAROTO que utilizam o multigrafo produzem caminhos primários com número médio de saltos semelhantes para qualquer carga simulada. Além disso, sob cargas baixas, os algoritmos RAMBO, STOP e FRSCA produzem caminhos com número de saltos semelhantes aos produzidos pelos algoritmos RAMBO, APPROVE e MAROTO, pois, sob estas cargas, os algoritmos com modulação fixa conseguem 100% de aceitação nas requisições. O algoritmo CaP-DPPM produz caminhos primários com o segundo maior número médio de saltos quando comparados com outros algoritmos, o que juntamente com a razão de fragmentação justifica o maior BBR gerado por ele.

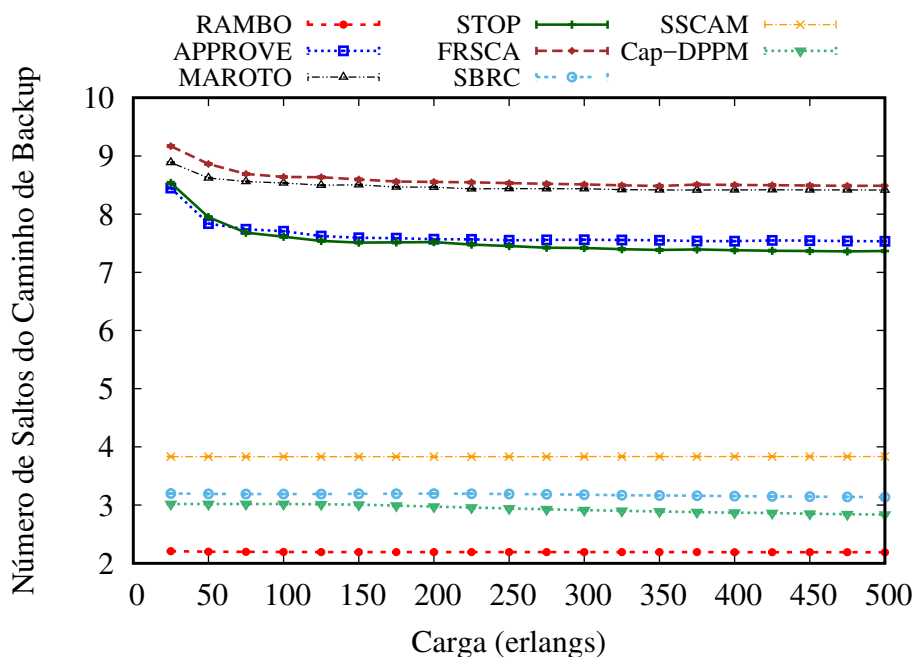


Figura 6.18: Número médio de saltos do caminho de *backup* dos algoritmos Cap-DPPM, SSCAM, SBRC, STOP, FRSCA, RAMBO, APPROVE e MAROTO em função da carga da rede para topologia NSF.

A Figura 6.18 mostra o número médio de saltos do caminho de *backup* gerado para a topologia NSF. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com duas vezes o número médio de saltos do que os caminhos que não utilizam. A menor conectividade da topologia NSF influenciou os tamanhos dos *p-cycles* criados de forma que os algoritmos MAROTO e FRSCA produziram caminhos com maior número médio de saltos que os produzidos pelos algoritmos STOP e APPROVE, uma vez que os algoritmos MAROTO e FRSCA não priorizam a criação de *p-cycles* que utilizam a menor quantidade de recursos, e sim a criação de *p-cycles* com interferência mínima nos caminhos primários. Por sua vez, os algoritmos STOP e APPROVE possui maior número médio de saltos que os algoritmos SSCAM, CaP-DPPM e RAMBO. O algoritmo SSCAM produz maior número de saltos do caminho de *backup* que o algoritmo CaP-DPPM para todas as cargas simuladas como consequência do algoritmo CaP-DPPM produzir maior bloqueio de requisições.

6.7 Síntese do capítulo

Neste capítulo, apresentaram-se os algoritmos RAMBO, APPROVE e MAROTO para proteção de redes ópticas elásticas com multiplexação espacial, utilizando agregação de tráfego e sobreposição de espectro e modulação adaptativa. Como nos capítulos anteriores, os algoritmos apresentados utilizaram uma técnica de multigrafo para simplificar o processo de encontrar caminhos através de topologias com múltiplos núcleos.

Os algoritmos apresentados são extensões dos algoritmos do Capítulo 5, que utilizam técnicas de modulação adaptativa para criação dos caminhos. Os valores de BBR produzidos pelos algoritmos propostos evidenciam o benefício de utilizar em conjunto a modulação adaptativa, agregação de tráfego e sobreposição de espectro. Os resultados apresentados mostraram que o uso destas técnicas em conjunto com *p-cycle* possibilitam resultados comparáveis aos gerados por algoritmos que usam caminhos de backup compartilhados, quando considerado topologias com baixa conectividade, como a topologia NSF, no que diz respeito ao bloqueio gerado por esses algoritmos.

Com a utilização das técnicas de modulação adaptativa, agregação de tráfego e sobreposição de espectro diminuiu-se o bloqueio das requisições. No entanto, a aceitação de um maior número de requisições também aumentou o CpS gerado por esses algoritmos.

Capítulo 7

Roteamento híbrido para proteção de redes ópticas elásticas com multiplexação espacial

Em redes ópticas WDM, a utilização de roteamento multicaminho tem como principal objetivo rotear aplicações com altos requisitos de banda, geralmente maiores do que a capacidade de um comprimento de onda. No entanto, para as redes ópticas elásticas o roteamento multicaminho é utilizado para aumentar a utilização dos recursos da rede. Adicionalmente este tipo de roteamento aumenta a confiabilidade da entrega.

O roteamento multicaminho visa explorar eficientemente os recursos da rede, utilizando vários caminhos entre pares origem-destino. O roteamento multicaminho tem potencial para agregar largura de banda em vários caminhos, permitindo que uma rede suporte taxas de transferência de dados superiores ao que é possível em um caminho único [16] e [105]. A utilização de roteamento multicaminhos em EONs pode oferecer vantagens de maior utilização do espectro, menor probabilidade de bloqueio de banda larga [81, 21, 49, 84], e incremento de sobrevivência [81, 84, 107], já que o volume total da demanda está dividido em várias partes (que são transmitidas por caminhos diferentes).

O passo inicial na criação de multicaminhos é calcular um conjunto de caminhos entre fonte e destino. Segundo [16], duas características podem ser utilizadas para determinar um conjunto de caminhos: a quantidade de caminhos, ou seja, o número de caminhos disponíveis entre os nós origem e destino e a independência do caminho; essa característica refere-se ao quão disjuntos são os caminhos que compõem o multicaminho.

Uma característica da utilização de roteamento multicaminhos é que a divisão do tráfego em múltiplos caminhos geram retardo diferencial, causado pelos diferentes retardos de propagação, o que pode acarretar a chegada de pacotes fora de ordem no nó de destino, requerendo memória para o re-sequenciamento desses pacotes [23]. Neste trabalho, assume-se que os nós destino possuem a quantidade de memória necessária para que as aplicações em tempo real não sejam afetadas.

Neste capítulo, propõe-se três algoritmos de proteção para redes ópticas elásticas com multiplexação espacial utilizando roteamento híbrido, isto é, roteamento monocaminho e multicaminho. Primeiramente, tenta-se encontrar caminhos simples, que satisfazem a requisição. Caso não seja possível utiliza-se roteamento multicaminho. Os algoritmos apresentados podem sele-

cionar um subconjunto de k , com $k = 2$ potenciais caminhos ópticos para atender a conexão solicitada. Além disso, os algoritmos apresentados estendem os algoritmos apresentados no capítulo 3, adicionando aos algoritmos roteamento multicaminho, quando monocaminhos não são possíveis.

Este capítulo está organizado conforme segue. Na Seção 7.1, apresenta-se uma revisão bibliográfica. Na seção 7.2 apresenta-se a notação utilizada nos algoritmos apresentados nesse capítulo. Nas Seções 7.3, 7.4, 7.5 e 5.6 descrevem-se, respectivamente, o algoritmo *Multipath rOuting For multicOre network* (MOFIO), o algoritmo *P-cycle path pRotection and multiPath routing for Multicore nEtwork* (PRIME) e o *Minimum InterfereNce and multiPath routing for MulticOre Networks* (MINION). Na Seção 7.6, compara-se o desempenho dos algoritmos propostos e de outros dois algoritmos. Finalmente, na Seção 7.7, apresenta-se a síntese do capítulo.

7.1 Trabalhos relacionados

Na literatura, existem diversos trabalhos que tratam de roteamento multicaminho para redes ópticas; no entanto, poucos trabalhos estudam a proteção destes caminhos.

Os autores em [31] propuseram um ILP e uma heurística de roteamento multicaminho e alocação de espectro para redes ópticas elásticas protegidas. O esquema proposto atinge uma melhora na utilização de recursos em comparação com os esquemas de proteção tradicionais. Além de multicaminho, os algoritmos utilizaram diferentes formatos de modulação.

Em [34], estuda-se a sobrevivência de redes ópticas elásticas através de roteamento multicaminho, permitindo a divisão de uma demanda em vários caminhos de roteamento, garantindo a realização de um volume de demanda específico no caso de uma falha de um único enlace. O artigo propõe uma ILP e uma heurística para resolver o problema de sobrevivência em redes ópticas elásticas utilizando multicaminho.

Chen *et.al* em [12] propuseram dois algoritmos de roteamento multicaminho e de atribuição de espectro (RSA) utilizando modulação adaptativa de acordo com a distância entre os nós finais para redes ópticas elásticas. Eles analisaram a eficácia dos algoritmos em cenários de tráfego dinâmico. Os autores mostraram que a combinação de roteamento multicaminho e modulação adaptativa em redes ópticas elásticas reduz a taxa de bloqueio em relação ao roteamento de caminho único.

Foram investigados em [22] os ganhos potenciais, empregando conjuntamente agregação de tráfego e multicaminho para redes ópticas elásticas. Através de simulações os autores demonstraram que a agregação de tráfego e roteamento multicaminho, juntos, aumentam a eficiência espectral e reduzem o consumo de recursos em relação aos esquemas existentes.

Os autores em [84] consideraram o impacto da fragmentação do espectro ao longo de caminhos simples e de multicaminho em redes ópticas elásticas. Mostrou-se que o esquema de multicaminho atinge maior eficiência espectral que o esquema de provisionamento com caminho único tradicional.

Apesar de alguns trabalhos já terem usado multicaminho para prover proteção em redes ópticas elásticas, nenhum trabalho tratou de proteção de redes ópticas elásticas com multiplexação espacial.

A Tabela 7.1 apresenta uma comparação entre os trabalhos relacionados e os algoritmos

proposto neste capítulo, os trabalhos foram categorizados com relação ao provisionamento de proteção, à utilização de redes ópticas elásticas com multiplexação espacial, a utilização de multicaminhos e por fim a utilização de *p-cycle* e SBPP para prover proteção.

Tabela 7.1: Comparação dos trabalhos relacionados com os algoritmos MOFIO, MINION e PRIME.

Abordagem	Proteção	SDM	Multicaminho	SBPP	P-cycle
Gao <i>et al.</i> [31]	Sim	Não	Sim	Sim	Não
Goscien <i>et al.</i> [34]	Sim	Não	Sim	Sim	Não
Chen <i>et al.</i> [12]	Não	Não	Sim	Não	Não
Dharmaweera <i>et al.</i> [22]	Não	Não	Sim	Não	Não
Ruan <i>et al.</i> [84]	Não	Não	Sim	Não	Não
MOFIO	Sim	Sim	Sim	Sim	Não
PRIME	Sim	Sim	Sim	Não	Sim
MINION	Sim	Sim	Sim	Não	Sim

Dentre todos os trabalho apresentados, apenas os introduzidos nesse capítulo, tratam de proteção e roteamento multicaminho em redes ópticas elásticas com multiplexação espacial.

7.2 Notação

A notação matemática a seguir será usada neste capítulo. A notação incrementa a notação do Capítulo 3. As principais diferenças encontram-se nos aspectos que tratam o multicaminho:

k : número de caminhos utilizados em cada requisição;

q : demanda de largura de banda d dividida pelo número de caminhos k , acrescida da banda de guarda;

A_n^k : sequência k de \tilde{G}_n tal que o nó fonte s é o menor nó ordenado e d é o maior nó ordenado;

$A = \{A_n^k\}$: conjunto de k -caminhos;

$R = \{R^k\}$: conjunto que guarda k -caminhos de *backup* temporariamente antes de estabelecer a conexão de todos os k caminhos;

$Y = \{Y^k\}$: conjunto que guarda k *p-cycles* temporariamente antes de estabelecer a conexão de todos os k caminhos;

$W(A_n^k)$: $\sum_{\tilde{e}_{\tilde{u},\tilde{v}} \in \{A_n^k\}} \tilde{e}_{\tilde{u},\tilde{v}}$: o peso do caminho A_n^k , a soma dos pesos de todas as arestas na sequência;

$W(A) = \{W(A_n^k)\}$: conjunto de todos os pesos dos k caminhos;

$W_{A_{s,d}^k}$ = peso dos k menores caminhos entre s e d ;

$W(R) = \{R^k\}$: conjunto de todos os pesos dos k -caminhos de *backup* selecionados;

W_{R^k} = peso dos k caminho de *backup* selecionado;

$\delta k(G, r(s, d, b))$: função que procura k -caminhos em todos os \tilde{G}_{n,b_m} , diferente da criação do *p-cycle* em que os dois caminhos necessitam utilizar o mesmo núcleo e os mesmos *slots*, esta função encontra k caminhos independente de núcleos e *slots*, ou seja, cada caminho pode ser encontrado em grafos separados utilizando o algoritmo de Dijkstra.

7.3 O algoritmo MOFIO

O algoritmo introduzido nesta seção, chamado *Multi/singlepath rOuting For multiCOr network* (MOFIO), garante 100% de proteção para todos os caminhos estabelecidos em uma rede óptica elásticas com multiplexação espacial. O algoritmo MOFIO calcula um caminho primário e um caminho de *backup* compartilhado para estabelecer a conexão. Constatada a impossibilidade de se criar caminhos simples para estabelecer a conexão, o algoritmo MOFIO calcula dois caminhos primários e dois caminhos de proteção para prover comunicação entre o par origem-destino. Os caminhos de proteção podem ser compartilhados desde que os caminhos primários protegidos sejam disjuntos. O algoritmo MOFIO estende o algoritmo 2 que encontra apenas caminhos simples.

No algoritmo MOFIO, as linhas 1 a 16 são baseadas no algoritmo SBPPMC. A diferença é que para o SBPPMC em caso de impossibilidade de encontrar um dos caminhos, a requisição é bloqueada, por outro lado, no algoritmo MOFIO, a requisição é dividida em multicaminhos, com a finalidade de balanceamento de carga, gerando melhor utilização dos recursos. O algoritmo MOFIO encontra o menor caminho (linha 2) para cada grafo gerado (linha 1). O menor caminho é escolhido na linha 3. Na linha 4, as arestas correspondentes ao caminho será marcada como ∞ . Na linha 6, um caminho que possa proteger o caminho primário é buscado nos caminhos de *backup* ativos. Na linha 9, um caminho de *backup* é criado nos casos em que não existem caminhos ativos. As linha 18 a 41 só serão executadas se os caminhos simples não for suficiente, sendo necessária uma abordagem multicaminho. A linha 18 gera os multigrafos para uma demanda de largura de banda q , que é a largura de banda b dividida pela quantidade de multicaminhos $q = b/k$. Na linha 19, são encontrados k multicaminhos a serem utilizados como caminhos primários para todos os grafos gerados. Na linha 20, são escolhidos os menores caminhos e seus pesos. Na linha 21, as arestas dos caminhos recebem peso ∞ . Nas linhas 22 a 36, procura-se um caminho de *backup* para cada caminho escolhido. O caminho de *backup* pode ser compartilhado (linha 24) ou ainda pode ser criado especificamente para a requisição (linha 27). Na linha 25, o caminho de *backup* é selecionado para ser estabelecido caso todos os outros caminhos de *backup* forem encontrados. Na linha 27, encontra-se um caminho de *backup* para o caminho primário em todos os grafos. Na linha 28, o menor caminho é escolhido. Se o peso do caminho for diferente de ∞ , o caminho de *backup* é escolhido e as arestas correspondentes tem seu peso alterado para o valor ∞ . A linha 36 garante que a requisição com multicaminhos somente seja aceita se todos os caminhos primários e de *backup* tenham pesos diferentes de ∞ . Caso contrario, a requisição é bloqueada na linha 39.

A complexidade do algoritmo MOFIO é analisada como: a complexidade de procurar caminhos simples mais a complexidade de procurar multicaminhos. A complexidade de procurar caminhos simples é $O(E + V \log V)$. A complexidade de procurar multicaminhos é analisada como: a complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar os caminhos primários, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para os caminhos de *backup*, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos. Logo, a complexidade de procurar multicaminhos é $(E + V) + 2 \times (C \times (N - b) \times (E + V \log V)) + E + V$. Como N , C e b podem ser expressados como constantes, a complexidade de procurar multicaminhos é $O(E + V \log V)$. E a complexidade do algoritmo MOFIO é $O((E + V \log V))$.

Algoritmo 15 MOFIO**Entrada:** $G, r(s, d, b), k = 2$ **Saída:** Caminho primário e Caminho de *backup*

```

1:  $\tau(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\tilde{G}_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} \neq \infty$  então
6:   Se  $\exists \nu(P_n, Z, r(s, d, b))$  então
7:     estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
8:   Senão
9:      $(W(Z_n), Z_n) = \beta(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
10:     $W_{Z_{s,d}} = W(Z_n) \mid \forall i W(Z_n) \leq W(Z_i)$ 
11:    Se  $W_{Z_{s,d}} \neq \infty$  então
12:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $Z_n$ 
13:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in Z_i$ 
14:    Fim Se
15:  Fim Se
16: Fim Se
17: Se  $W_{P_{s,d}} = \infty \wedge (\nexists \nu(P_n, Z, r(s, d, b)) \vee W_{Z_{s,d}} = \infty)$  então
18:   $\tau(G, C, q)$ 
19:   $(W(A), A) = \delta k(\tilde{G}_{n,b}, r(s, d, q))$ 
20:   $W_{A_{s,d}^k} = W(A_n^k) \mid \forall i W(A_n^k) \leq W(A_i^k)$ 
21:   $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in A_i^k$ 
22:  Para todo  $A_n^k \in A$  faça
23:    Se  $W(A_n^k) \neq \infty$  então
24:      Se  $\exists \nu(A_n^k, Z, r(s, d, b))$  então
25:         $R^k = Z_n$ 
26:      Senão
27:         $(W(Z_n), Z_n) = \beta(\tilde{G}_{n,b}, A_n^k, r(s, d, b)) \quad \forall n \in \sigma$ 
28:         $W_{Z_{s,d}} = W(Z_n) \mid \forall i W(Z_n) \leq W(Z_i)$ 
29:        Se  $W_{Z_{s,d}} \neq \infty$  então
30:           $R^k = Z_n$ 
31:           $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in Z_i$ 
32:        Fim Se
33:      Fim Se
34:    Fim Se
35:  Fim Para todo
36:  Se  $W(A_n^k) \neq \infty \wedge W(R^k) \neq \infty \quad \forall k$  então
37:    estabelece requisição  $(r(s, d, b))$  como  $A$  e  $R$ 
38:  Senão
39:    bloqueia requisição  $(r(s, d, b))$ 
40:  Fim Se
41: Fim Se

```

7.4 O algoritmo PRIME

O algoritmo introduzido nesta seção, chamado *P-cycle path pProtection and hybrId routing for Multicore nEtwork* (PRIME), é uma extensão do algoritmo FIPPMC. Como no FIPPMC ele garante 100% de proteção para todas as requisições estabelecidas. Este algoritmo provê multi-

caminhos no estabelecimento de uma conexão no caso da impossibilidade de estabelecimento de caminhos simples apenas. O algoritmo PRIME como o algoritmo 3 utiliza *p-cycle* FIPP na proteção de caminhos. Nos casos em que a conexão utiliza um caminho primário simples, um *p-cycle* FIPP, que pode ser compartilhado, é utilizado como caminho de backup. Nos casos em que a conexão utiliza roteamento multicaminho, para cada caminho primário é estabelecido um *p-cycle* FIPP para proteção. Em nenhuma hipótese, o mesmo *p-cycle* pode proteger mais de um caminho da mesma conexão.

O algoritmo PRIME repete os passos realizados no algoritmo FIPPMC nas linhas 1 a 16. O algoritmo calcula os menores caminhos para os grafos gerados (linha 2). O menor caminho é encontrado na linha 3. Se um caminho for encontrado o algoritmo encontra dentre o *p-cycles* estabelecidos um *p-cycle* FIPP que protege o caminho primário (linha 6). Caso não exista nenhum *p-cycle* FIPP ativo que possa proteger a requisição, então um *p-cycle* FIPP é criado (linha 9). Caso o caminho primário ou o *p-cycle* FIPP não possa ser criado, ao contrario do algoritmo FIPPMC que bloqueia a requisição nos casos em que caminhos simples não podem ser estabelecidos, o algoritmo PRIME calcula multicaminhos para o estabelecimento da conexão. O algoritmo PRIME utiliza $k=2$ quando necessário utilizar multicaminho. A utilização de multicaminho através das linha 18 a 41, só serão executadas se e somente se os caminhos simples não forem suficientes. Na linha 18, multigrafos são gerados utilizando como parâmetro uma demanda de largura de banda q . Na linha 19, são encontrados k multicaminhos à serem utilizados como caminhos primários para todos os grafos gerados. Na linha 20, são escolhidos os menores caminhos e seus pesos. Na linha 21, as arestas dos caminhos recebem ∞ . As linhas 17 a 21 que realizam a escolha de k caminhos primários, é similar as mesmas linhas do algoritmo MOFIO, indicando a similaridade da criação de multicaminhos nos algoritmos. Nas linhas 22 a 36, procura-se um *p-cycle* FIPP, para ser usado como caminho de *backup* para cada caminho escolhido. O *p-cycle* FIPP pode ser compartilhado (linha 24), ou ainda pode ser criado especificamente para requisição (linha 27) e compartilhado por novas requisições. Na linha 25, o *p-cycle* FIPP é guardado para ser estabelecido caso todos os outros multicaminhos e *p-cycles* forem encontrados. Na linha 27, encontra-se um caminho de *backup* para o caminho primário em todos os grafos. Na linha 28, o menor caminho é escolhido. Se o peso do caminho for diferente de ∞ , o caminho de *backup* é guardado e arestas correspondentes tem seu peso alterado para o valor ∞ . A linha 36 garante que a requisição com multicaminhos somente seja aceita se todos os caminhos primários e de *backup* tiverem pesos diferentes de ∞ . Caso contrario, a requisição é bloqueada na linha 39.

A complexidade do algoritmo PRIME é analisada como segue: a complexidade de procurar caminhos simples mais a complexidade de procurar multicaminhos. A complexidade de procurar caminhos simples (caminho primário e *p-cycle*) é $O(E + V \log V)$. A complexidade de procurar multicaminhos é analisada como: a complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar os caminhos primários, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para os *p-cycles*, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos. Logo, a complexidade de procurar multicaminhos é $(E + V) + 2 \times (C \times (N - b) \times (E + V \log V)) + E + V$. Como N , C e b podem ser expressados como constantes, a complexidade de procurar multicaminhos é $O(E + V \log V)$. E a complexidade do algoritmo MOFIO é $O((E + V \log V) + (E + V \log V)) = O(E + V \log V)$.

Algoritmo 16 PRIME**Entrada:** $G, r(s, d, b), k = 2$ **Saída:** Caminho primário e Caminho de backup

```

1:  $\tau(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\tilde{G}_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) | \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} \neq \infty$  então
6:   Se  $\exists \mu(P_n, T, r(s, d, b))$  então
7:     estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
8:   Senão
9:      $(W(T_n), T_n) = \xi(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
10:     $W_{T_{s,d}} = W(T_n) | \forall i W(T_n) \leq W(T_i)$ 
11:    Se  $W_{T_{s,d}} \neq \infty$  então
12:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
13:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
14:    Fim Se
15:  Fim Se
16: Fim Se
17: Se  $W_{P_{s,d}} = \infty \wedge (\nexists \mu(P_n, T, r(s, d, b)) \vee W_{T_{s,d}} = \infty)$  então
18:   $\tau(G, C, q)$ 
19:   $(W(A), A) = \delta k(\tilde{G}_{n,b}, r(s, d, q))$ 
20:   $W_{A_{s,d}^k} = W(A_n^k) | \forall i W(A_n^k) \leq W(A_i^k)$ 
21:   $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in A_i^k$ 
22:  Para todo  $A_n^k \in P$  faça
23:    Se  $W(A_n^k) \neq \infty$  então
24:      Se  $\exists \mu(A_n^k, T, r(s, d, b))$  então
25:         $Y^k = T_n$ 
26:      Senão
27:         $(W(T_n), T_n) = \xi(\tilde{G}_{n,b}, A_n^k, r(s, d, b)) \quad \forall n \in \sigma$ 
28:         $W_{T_{s,d}} = W(T_n) | \forall i W(T_n) \leq W(T_i)$ 
29:        Se  $W_{T_{s,d}} \neq \infty$  então
30:           $Y^k = T_n$ 
31:           $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
32:        Fim Se
33:      Fim Se
34:    Fim Se
35:  Fim Para todo
36:  Se  $W(A_n^k) \neq \infty \wedge W(Y^k) \neq \infty \quad \forall k$  então
37:    estabelece requisição  $(r(s, d, b))$  como  $A$  e  $Y$ 
38:  Senão
39:    bloqueia requisição  $(r(s, d, b))$ 
40:  Fim Se
41: Fim Se

```

7.5 O algoritmo MINION

O algoritmo introduzido nesta seção, chamado *Minimum Interference and hybrid routing for Multicore Networks* (MINION) estende o algoritmo 4. Similarmente aos outros algoritmos desta Tese, o algoritmo MINION aceita apenas requisições que possam ser protegidas. Dife-

rentemente do algoritmo PRIME que utiliza *p-cycles* FIPP para prover proteção aos caminhos primários, o algoritmo MINION adicionalmente utiliza interferência mínima na criação do *p-cycle*, priorizando *p-cycles* que utilizam enlaces disjuntos do caminho protegido (*p-cycles* transzonais). O algoritmo prioriza, também, estabelecer conexões através de caminhos simples. No entanto, quando estes não são possíveis, o estabelecimento de conexões pode ocorrer através de roteamento multicaminho.

O algoritmo MINION repete os passos realizados no algoritmo MIFMC nas linhas 1 a 20. Nestas linhas, o algoritmo MINION calcula um menor caminho utilizando o mecanismo de grafos apresentado no Capítulo 1 e procura *p-cycles* que possa ser compartilhados para proteger a requisição. Caso não seja possível, o algoritmo calcula um *p-cycle* FIPP transzonal ao caminho encontrado (linha 9). Caso não seja possível, ele calcula um *p-cycle* FIPP qualquer (linha 12). Dada inexistência de caminhos simples ou *p-cycle* FIPP para estabelecer a requisição (linha 21), o algoritmo MINION calcula através de roteamento multicaminho, com $k=2$, caminhos e *p-cycles* FIPP que possam estabelecer. A utilização de multicaminho nas linha 21 a 49, só serão executadas se os caminhos simples não forem suficientes. Na linha 22, multigrafos são gerados utilizando como parâmetro uma demanda de largura de banda q . Na linha 23, são encontrados k caminhos à serem utilizados como caminhos primários para todos os grafos gerados. Na linha 24, são escolhidos os menores caminhos e seus pesos. Na linha 25, os pesos das arestas dos caminhos recebem o valor ∞ . As linhas 22 a 25, que realizam a escolha de k caminhos primários, são similar as correspondentes do algoritmo MOFIO e PRIME, indicando a similaridade da criação de multicaminhos nos algoritmos. Nas linhas 26 a 43 procura-se um *p-cycle* FIPP, para ser usado como caminho de *backup* compartilhado para cada caminho escolhido. Na linha 28, para cada caminho primário selecionado, busca-se um *p-cycle* dentre os *p-cycles* FIPP ativo para proteger o caminho. Caso não seja possível encontrar um *p-cycle* ativo para proteger o caminho primário, calcula-se na linha 31 um *p-cycle* transzonal ao caminho primário para protegê-lo em todos os grafos gerados. Se o peso do menor *p-cycle* FIPP transzonal ao caminho primário (linha 32) for ∞ , um *p-cycle* FIPP que pode utilizar os enlaces do *p-cycle* é calculado para todos grafos gerados (linha 35). Se todos os *p-cycle* FIPPs buscados (linha 28) e criados (linha 32 e 36) tiverem peso ∞ a requisição é bloqueada (linha 47), caso contrário a requisição é aceita (linha 45).

A complexidade do algoritmo MINION é analisada como: a complexidade de procurar caminhos simples mais a complexidade de procurar multicaminhos. A complexidade de procurar caminhos simples (caminho primário e *p-cycle*) é $O(E + V \log V)$. A complexidade de procurar multicaminhos é analisada como: a complexidade de transformar o multigrafo em grafos é $O(E + V)$. Para encontrar os caminhos primários, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos, em que a complexidade amortizada de Dijkstra é $O(E + V \log V)$. Da mesma forma, para os *p-cycles*, o algoritmo de Dijkstra é executado em $C \times (N - b)$ grafos. Logo, a complexidade de procurar multicaminhos é $(E + V) + 2 \times (C \times (N - b) \times (E + V \log V)) + E + V$. Como N , C e b podem ser expressados como constantes, a complexidade de procurar multicaminhos é $O(E + V \log V)$. E a complexidade do algoritmo MOFIO é $O((E + V \log V) + (E + V \log V)) = O(E + V \log V)$.

Algoritmo 17 MINION**Entrada:** $G, r(s, d, b), k = 2$ **Saída:** Caminho primário e Caminho de backup

```

1:  $\tau(G, C, b)$ 
2:  $(W(P_n), P_n) = \delta(\tilde{G}_{n,b}, r(s, d, b)) \quad \forall n \in \sigma$ 
3:  $W_{P_{s,d}} = W(P_n) \mid \forall i W(P_n) \leq W(P_i)$ 
4:  $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in P_i$ 
5: Se  $W_{P_{s,d}} \neq \infty$  então
6:   Se  $\exists \mu(P_n, T, r(s, d, b))$  então
7:     estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
8:   Senão
9:      $(W(T_n), T_n) = \vartheta(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
10:     $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
11:    Se  $W_{T_{s,d}} = \infty$  então
12:       $(W(T_n), T_n) = \xi(\tilde{G}_{n,b}, P_n, r(s, d, b)) \quad \forall n \in \sigma$ 
13:       $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
14:    Fim Se
15:    Se  $W_{T_{s,d}} \neq \infty$  então
16:      estabelece requisição  $(r(s, d, b))$  como  $P_n$  e  $T_n$ 
17:       $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
18:    Fim Se
19:  Fim Se
20: Fim Se
21: Se  $W_{P_{s,d}} = \infty \wedge (\nexists \mu(P_n, T, r(s, d, b)) \vee W_{T_{s,d}} = \infty)$  então
22:    $\tau(G, C, q)$ 
23:    $(W(A), A) = \delta k(\tilde{G}_{n,b}, r(s, d, q))$ 
24:    $W_{A_{s,d}^k} = W(A_n^k) \mid \forall i W(A_n^k) \leq W(A_i^k)$ 
25:    $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in A_i^k$ 
26:   Para todo  $A_n^k \in P$  faça
27:     Se  $W(A_n^k) \neq \infty$  então
28:       Se  $\exists \mu(A_n^k, T, r(s, d, b))$  então
29:          $Y^k = T_n$ 
30:       Senão
31:          $(W(T_n), T_n) = \vartheta(\tilde{G}_{n,b}, A_n^k, r(s, d, b)) \quad \forall n \in \sigma$ 
32:          $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
33:       Fim Se
34:       Se  $W_{T_{s,d}} = \infty$  então
35:          $(W(T_n), T_n) = \xi(\tilde{G}_{n,b}, A_n^k, r(s, d, b)) \quad \forall n \in \sigma$ 
36:          $W_{T_{s,d}} = W(T_n) \mid \forall i W(T_n) \leq W(T_i)$ 
37:       Fim Se
38:       Se  $W_{T_{s,d}} \neq \infty$  então
39:          $Y^k = T_n$ 
40:          $w(e_{u,v,i}) = \infty \quad \forall \{u, v\} \in T_i$ 
41:       Fim Se
42:     Fim Se
43:   Fim Para todo
44:   Se  $W(A_n^k) \neq \infty \wedge Y^k \neq \infty \quad \forall k$  então
45:     estabelece requisição  $(r(s, d, b))$  como  $A$  e  $Y$ 
46:   Senão
47:     bloqueia requisição  $(r(s, d, b))$ 
48:   Fim Se
49: Fim Se

```

7.6 Resultados numéricos

Neste capítulo, avaliam-se os benefícios da utilização de roteamento híbrido através dos algoritmos MOFIO, PRIME e MINION. Os algoritmos calculam caminhos simples para caminho primário e de backup. Caso não seja possível a utilização de caminhos simples, os algoritmos calculam multicaminho. Utilizou-se o valor de $k = 2$, que representa a quantidade de caminhos utilizados. Os algoritmos utilizam técnicas específicas para proteção, dando continuidade aos algoritmos gerados desde o capítulo 3.

Para avaliar o desempenho dos algoritmos proposto neste capítulo, como nos capítulos anteriores experimentos de simulação foram realizados utilizando o simulador FlexGridSim [59]. Em cada simulação, 100.000 requisições foram geradas e utilizou-se o método de replicação independente e adotou-se nível de confiança de 95% para os intervalos de confiança. Diferentes topologias com diferentes conectividades foram empregadas nas simulações, a Pan-European (Figura 3.2(a)), a USA (Figura 3.2(b)) e a NSF (Figura 3.2(c)). O espectro foi dividido em 320 *slots* de espectro e as fibras simulada possuem 7 núcleos. A carga foi variada entre 25 e 500 erlangs em intervalos de 25 erlangs. Sete tipos de requisições foram utilizadas 25 Gbps, 50 Gbps, 125 Gbps, 200 Gbps, 500 Gbps, 750 Gbps e 1 Tbps. Para comparação dos algoritmos proposto neste capítulo foram utilizados os algoritmos Cap-DPP e SSCA já analisados no capítulo 3. Além disso, os algoritmo FIPPMC, SBPPMC e MIFMC, que utilizam roteamento simples foram utilizados para comparação com os novos algoritmos que utilizam roteamento híbrido. Através desta comparação é possível avaliar o impacto de utilizar roteamento multicaminho quando caminhos simples não são possíveis.

As métricas utilizadas foram as mesmas utilizadas nos capítulos anteriores. No entanto, o número médio de saltos é contabilizado por conexão, ao invés de contabilizar por caminho, ou seja, é somado todos os saltos de todos os caminhos de uma dada conexão.

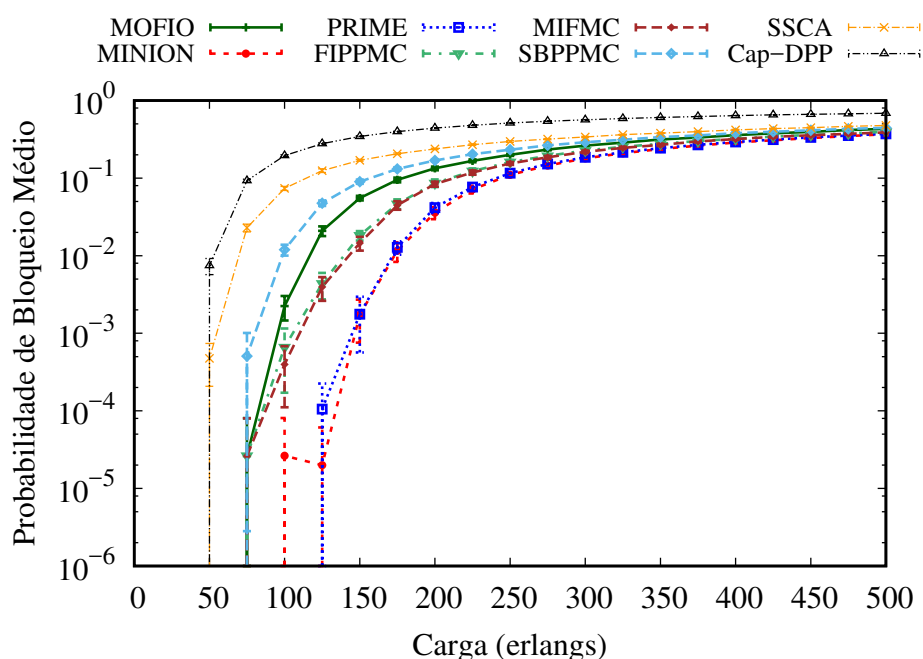


Figura 7.1: Bloqueio de banda dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.

A Figura 7.1 mostra a probabilidade de bloqueio para a topologia Pan-European. O algoritmo CaP-DPP e SSCA iniciam o bloqueio de requisições sob cargas de 50 erlangs. Os algoritmos SBPPMC, FIPPMC e MIFMC, que utilizam o multigrafo e apenas caminhos simples, iniciam o bloqueio de requisições sob cargas de 75 erlangs. Os algoritmos PRIME, MOFIO e MINION, que utilizam roteamento multicaminho aumentam a possibilidade de utilização e estabelecimento de caminhos ópticos através de múltiplas rotas e, conseqüentemente, aumenta as chances de encontrar recursos disponíveis para suprir a banda passante demandada pelas requisições de conexão, produzindo, portanto, valores menores de BBR que quando utilizado caminhos simples. O algoritmo MOFIO iniciou o bloqueio de requisições sob cargas de 75 erlangs, produzindo maiores valores de BBR que os algoritmos que utilizam *p-cycle* e caminhos simples. Os algoritmos PRIME e MINION iniciaram o bloqueio de requisições apenas sob carga de 125 e 100 erlangs, respectivamente. Apesar do algoritmo PRIME iniciar o bloqueio sob cargas maiores que o MINION, entre cargas de 125 erlangs a 150 erlangs o algoritmo MINION produz um menor bloqueio. Os algoritmos CaP-DPP e SSCA que utilizam apenas caminhos simples, produziram maior bloqueio evidenciando os benefícios de utilizar roteamento multicaminho nos casos em que caminhos simples não são possíveis. Os algoritmos FIPPMC, MIFMC, MINION e PRIME, que utilizam *p-cycle* FIPP, beneficiaram-se da maior conectividade da topologia Pan-European produzindo valores de BBR menores que aqueles valores gerados pelos algoritmos SBPPMC e MOFIOs. Apesar do maior uso de recursos da algoritmo MINION para criação do caminho de backup, o aumento do número de caminhos com o uso de roteamento multicaminho, produz melhor oportunidade de compartilhamento, produzindo valores de BBR menores.

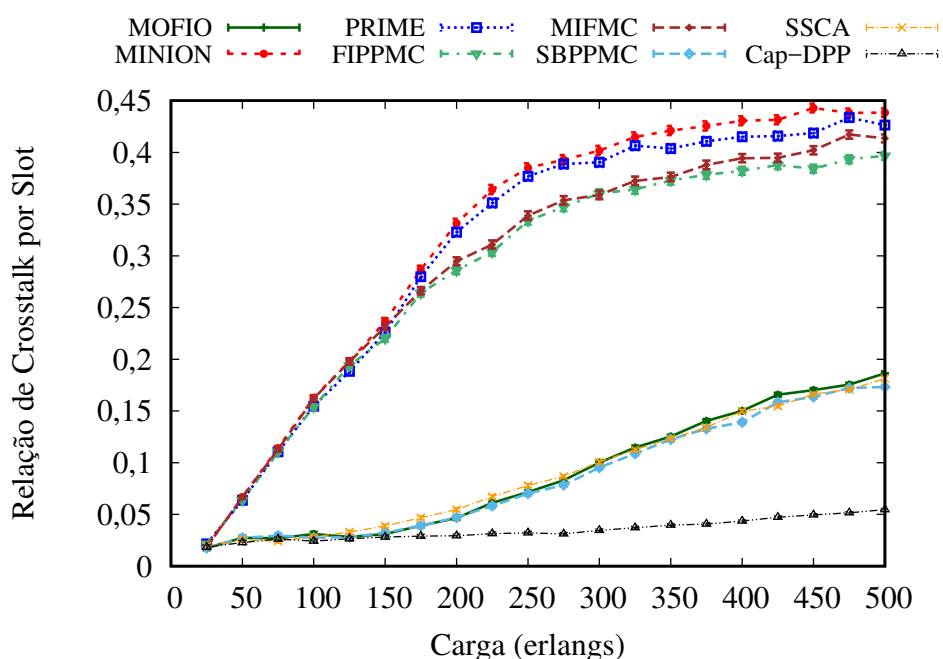


Figura 7.2: *Crosstalk* por *slot* dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.

A Figura 7.2 mostra a relação de *crosstalk* por *slot* gerado para a topologia Pan-European. Todos os algoritmos iniciam em um valor de 0,02. Os algoritmos FIPPMC, MIFMC, PRIME

e MINION, que utilizam *p-cycle* FIPP, produzem valores de CpS que aumentam rapidamente chegando a 0,40, 0,42, 0,43 e 0,44, respectivamente. O CpS gerado pelo algoritmo SBPPMC aumenta até 0,17 enquanto que os algoritmos MOFIO e SSCA aumentam até 0,18. Contudo, o algoritmo MOFIO aceita um número maior de requisições. O algoritmo CaP-DPP aumenta até 0,05, produzindo um CpS baixo em relação aos outros algoritmos devido ao menor número de requisições aceitas, o que implica em um maior uso de recursos. Os valores de CpS gerados pelos algoritmos FIPPMC, MIFMC, MINION e PRIME possuem valores mais altos que outros algoritmos, isto acontece pois mais requisições são aceitas e além disso *p-cycles* FIPP utilizam mais recursos para proteção, que pode implicar em caminhos primários com maior número médio de saltos e consequentemente mais *slots* sofrem com CpS.

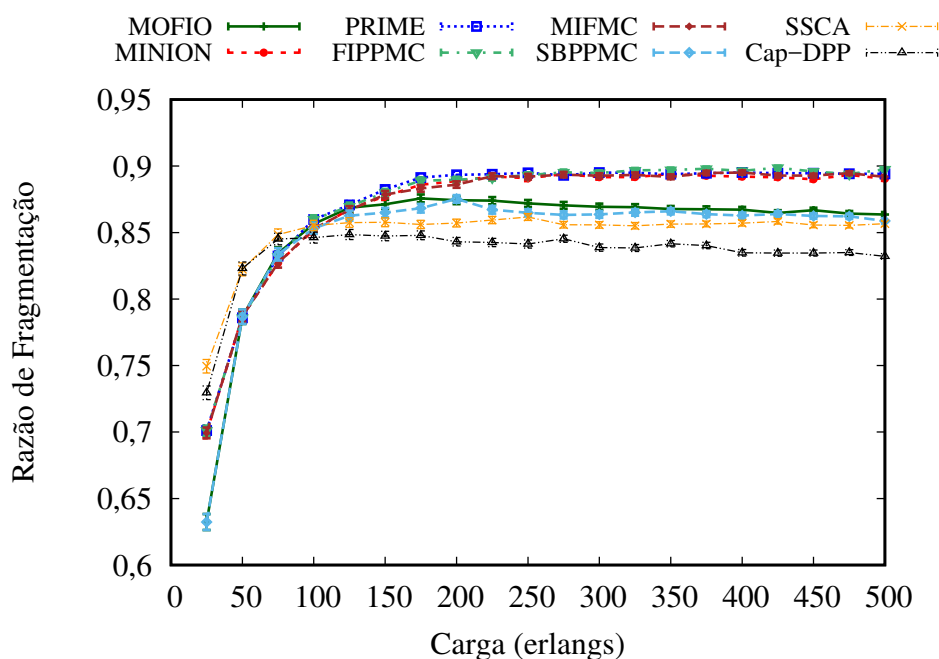


Figura 7.3: Razão de fragmentação dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.

A Figura 7.3 mostra a razão de fragmentação gerada para a topologia Pan-European. Sob altas cargas, o algoritmo PRIME seguido dos algoritmos FIPPMC, MINION e MIFMC, respectivamente, produzem as maiores relações de fragmentação, pois os algoritmos que utilizam *p-cycle* FIPP utilizam mais recursos de backup, apesar dos recursos não serem utilizados, a reserva gera fragmentação. O algoritmo MINION produz caminhos de *backup* com maior número de saltos que os caminhos de *backup* produzidos pelo algoritmo PRIME. No entanto, o algoritmo MINION produz maior compartilhamento o que faz com que menos caminhos de *backup* sejam estabelecidos gerando menor razão de fragmentação que o algoritmo PRIME. Apesar do algoritmo CaP-DPP produzir os menores valores de BBR, ele produz maior razão de fragmentação que o algoritmo SSCA, isto ocorre devido ao compartilhamento de recursos do caminho de *backup* do algoritmo SSCA gerando menor estabelecimento de caminhos de backup, consequentemente menor razão fragmentação. O algoritmo MOFIO produz menor razão de fragmentação que os algoritmos que utilizam *p-cycle* FIPP, pois um menor número de requisições são aceitas. Os algoritmos que utilizam roteamento híbrido produzem razão de

fragmentação semelhante aqueles que utilizam caminhos simples, apesar dos algoritmos que utilizam roteamento híbrido produzirem menores valores de BBR.

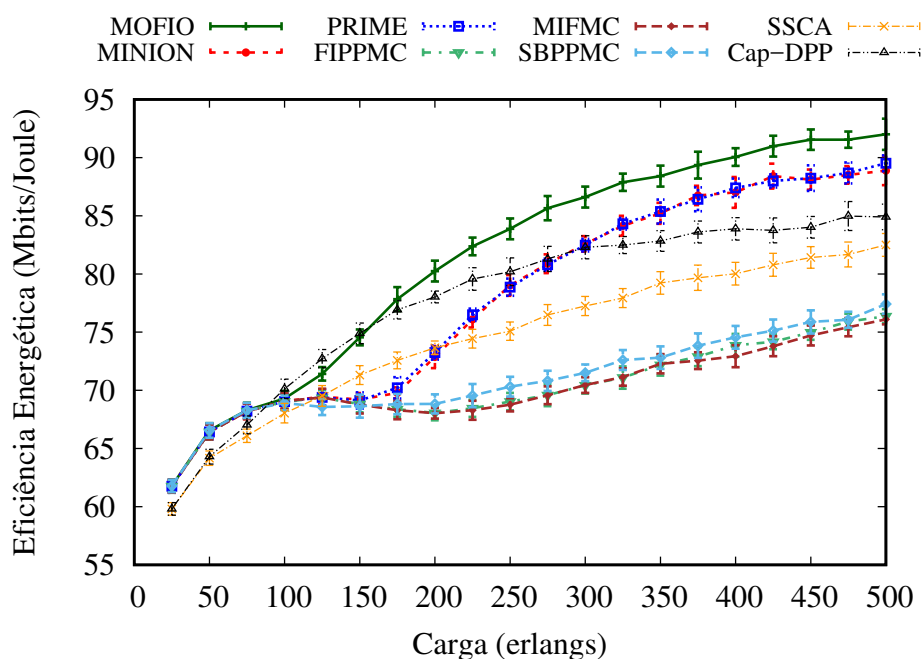


Figura 7.4: Eficiência Energética dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.

A Figura 7.4 mostra a eficiência energética gerada para a topologia Pan-European. A eficiência energética em conexões que utilizam multicaminhos, diferentemente das que utilizam monocaminho, leva em consideração não somente o número de saltos como também a banda que cada um dos caminhos aloca. Além disso, ao dividir a requisição em multicaminhos é necessária o uso de banda de guarda para cada caminho utilizado, o que pode aumentar o consumo energético. Por outro lado, se ao dividir a requisição em multicaminhos os caminhos escolhidos forem menores, a eficiência energética é maior. Para cargas maiores que 150 erlangs, o algoritmo MOFIO produz a maior eficiência energética, o que indica que uma melhor utilização de recursos para um menor número de saltos por conexão. Os algoritmos PRIME e MINION produziram menor eficiência energética do que o algoritmo MOFIO, devido a maior BBR gerada pelo algoritmo MOFIO, o que faz com o algoritmo produza apenas caminhos mais curtos, e consequentemente maior eficiência energética. Os algoritmos que utilizam monocaminho geram menor eficiência energética do que aqueles que utilizam multicaminho principalmente sob altas cargas. Apesar do algoritmo SSCA produzir um alto BBR, ele produz uma baixa eficiência energética, como consequência de uma pior escolha dos caminhos a serem utilizados.

A Figura 7.5 mostra o número médio de saltos dos caminhos primários por requisição para a topologia Pan-European. Sob baixas cargas, os algoritmos SSCA e CaP-DPP produzem maior número de saltos, pois estes algoritmos não utilizam boas escolhas de rotas, ao contrário dos algoritmos SBPPMC, MIFMC, FIPPMC, MOFIO, PRIME e MINION que utilizam a técnica de multigrafos proposta nesta Tese. Sob baixas cargas, os algoritmos que utilizam roteamento híbrido produzem caminhos primários com número médio de saltos similares àqueles que utilizam roteamento monocaminho. Sob altas cargas os algoritmos que utilizam roteamento híbrido

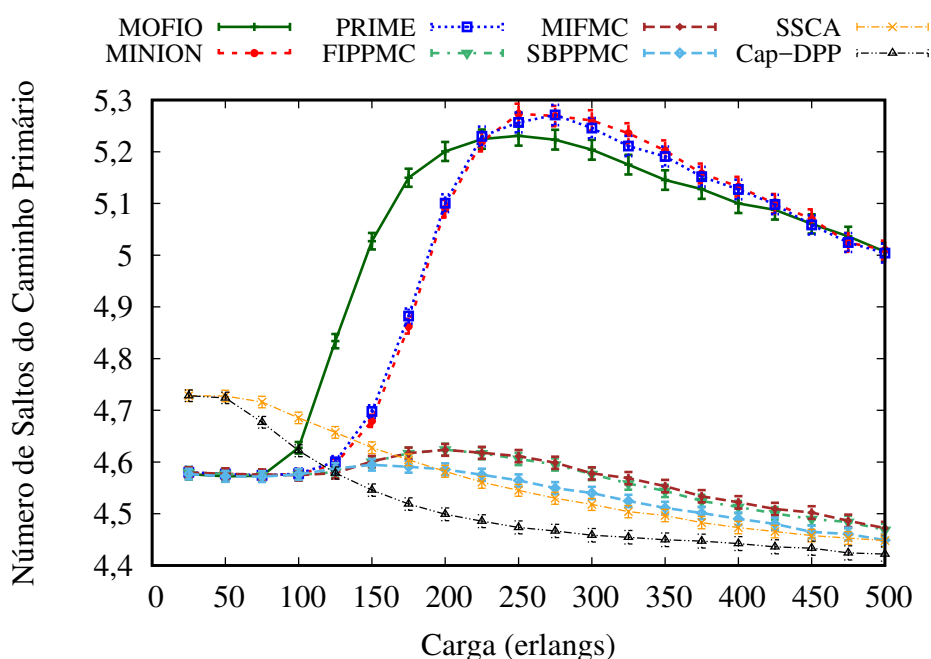


Figura 7.5: Número médio de saltos dos caminhos primários por requisição dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia Pan-European.

produzem um alto número médio de saltos, como consequência da possibilidade de mais de um caminho ser utilizado. A utilização de roteamento multicaminho não necessariamente implica em um maior uso de recursos, um vez que cada caminho necessita de aproximadamente a metade dos recursos. Sob altas cargas os algoritmos CaP-DPP e SSCA produzem altos BBRs com isso ocorre o decremento de número de saltos já que apenas menores caminhos são estabelecidos. Sob altas cargas os algoritmos FIPPMC, SBPPMC e MIFMC produzem caminhos com menores número médio de saltos dos caminhos primários por requisição que os algoritmos MOFIO, MINION e PRIME, devido ao uso de mais de um caminho.

A Figura 7.6 mostra o número médio de saltos dos caminhos de *backup* por requisição gerados para a topologia Pan-European. O algoritmo MINION produz o maior número médio de saltos do caminho de backup, pois além do uso de roteamento multicaminho este algoritmo prioriza encontrar ciclos disjuntos do caminho primário para formação de *p-cycles*. O algoritmo PRIME produz o segundo maior número médio de saltos do caminho de backup, influenciado também pelo roteamento multicaminho e pela necessidade de criação de ciclos para formar o *p-cycle*. Sob cargas altas, os algoritmos que utilizam roteamento híbrido produzem caminhos com maiores número de saltos que os algoritmos correspondentes que utilizam roteamento simples, pois com o aumento da carga torna-se necessário a utilização de roteamento multicaminhos, produzindo caminhos com maiores número médio de saltos.

A Figura 7.7 mostra a probabilidade de bloqueio para a topologia USA. Os algoritmos CaP-DPP e SSCA iniciaram o bloqueio de requisições sob carga de 50 erlangs. Os algoritmos FIPPMC e MIFMC, que utilizam *p-cycle* FIPP e roteamento simples, iniciaram o bloqueio de requisições sob carga de 125 erlangs, por sua vez o algoritmo SBPPMC, que utiliza roteamento simples e caminhos compartilhados para proteção, iniciou o bloqueio de requisições sob carga

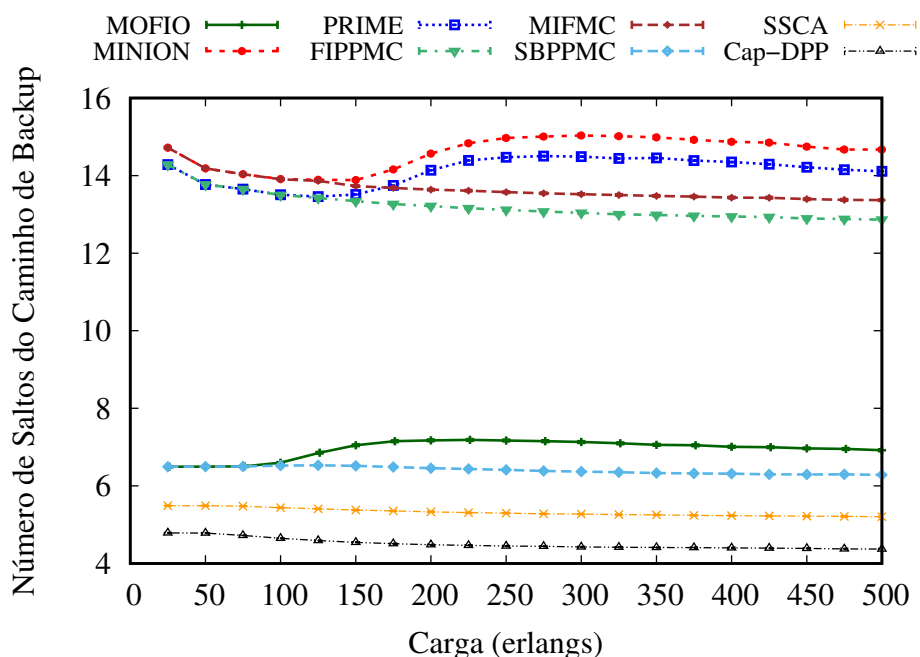


Figura 7.6: Número médio de saltos dos caminhos de *backup* dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, Pan-European.

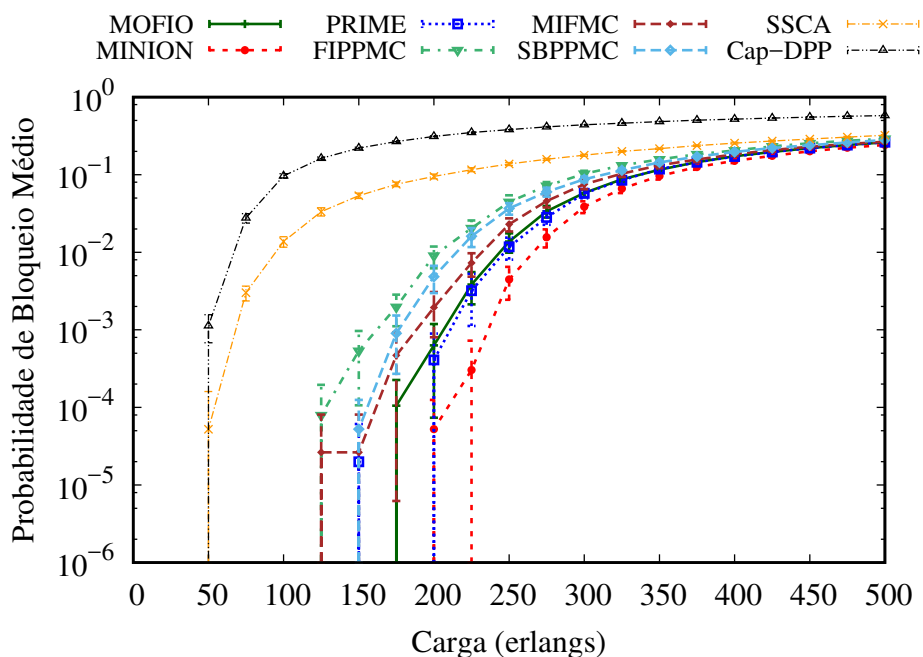


Figura 7.7: Bloqueio de banda dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.

de 150 erlangs. O algoritmo MOFIO, que utiliza roteamento híbrido, iniciou o bloqueio de requisições apenas sob cargas de 175 erlangs. Os algoritmos PRIME e MINION iniciaram o bloqueio de requisições apenas sob carga de 150 e 200 erlangs, respectivamente. Os algoritmos que utilizam apenas roteamento monocaminho produzem valores de BBR maiores que os seus respectivos algoritmos que utilizam roteamento híbrido. O algoritmo CaP-DPP produz valores

de BBR maiores que os valores de BBR produzidos pelo algoritmo SSCA, como consequência do compartilhamento de recursos de *backup* do algoritmo SSCA, utilizando menos recurso para proteção. O algoritmo MINION produz os menores valores de BBR, evidenciando a influência da utilização conjunta de *p-cycles*, roteamento multicaminho e interferência mínima na criação de caminhos de backup. O algoritmo MINION produz alto compartilhamento dos caminhos de backup, visto que ele produz caminhos de *backup* maiores. Sob cargas maiores que 200 erlangs os algoritmos PRIME e MOFIO produzem valores de BBR similares, isto ocorre pois a possibilidade de dividir a conexão para caminhos que utilizam *p-cycle*, facilita o compartilhamento, tirando mais vantagens do roteamento multicaminho que o algoritmo MOFIO.

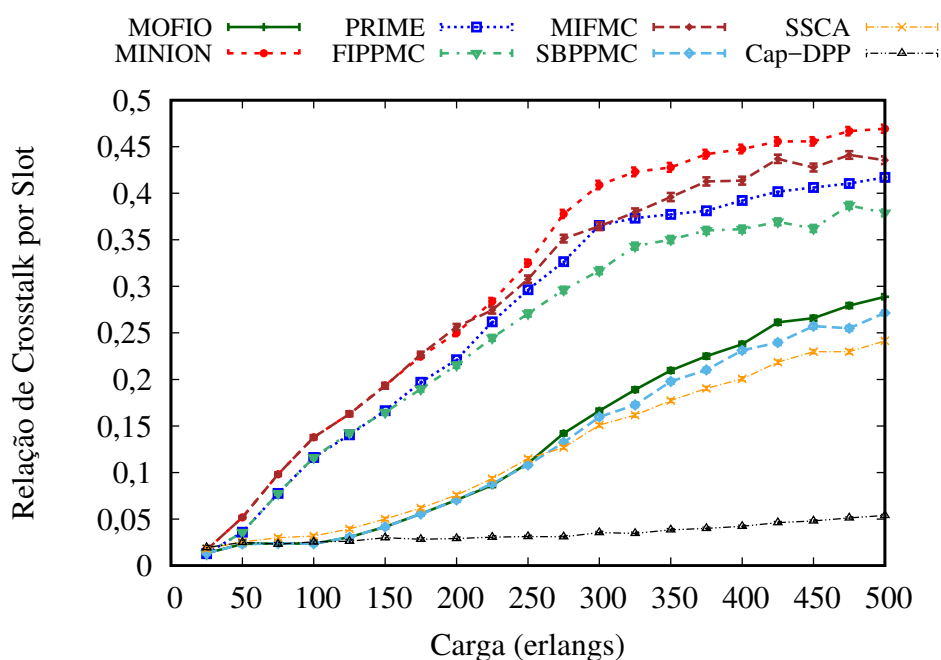


Figura 7.8: *Crosstalk* por *slot* dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.

A Figura 7.8 mostra a relação de *crosstalk* por *slot* gerado para a topologia USA. Os valores de CpS gerados pelos algoritmos FIPPMC, PRIME, MIFMC e MINION iniciam com um valor de 0,02 e aumentam até 0,38, 0,41, 0,43 e 0,47, respectivamente. Os valores de CpS gerados pelo o algoritmo SSCA, SBPPMC e MOFIO iniciam com um valor de 0,02 e aumentam até 0,24, 0,27 e 0,29, respectivamente. O algoritmo CaP-DPP produz o menor CPS, influenciado pelo menor número de requisições aceitas. Os algoritmos que utilizam *p-cycle* FIPP produzem os maiores valores de CpS, como consequência da maior utilização de recursos de *backup*, o que aumenta o número de saltos dos caminhos primários produzidos, e consequentemente aumentam o número de *slots* que sofrem com *crosstalk*. O algoritmo MOFIO apesar de rejeitar um número baixo de requisições produz CpS similar ao algoritmo SSCA que produz alto número de rejeições, isto ocorre devido a melhor escolha de caminhos e a utilização de roteamento multicaminho. Apesar dos algoritmos que utilizam roteamento híbrido produzirem valores de BBR menores, os algoritmos que utilizam roteamento simples produzem valores de CpS próximos aos dos gerados pelos correspondentes algoritmos que utilizam roteamento híbrido.

A Figura 7.9 mostra a razão de fragmentação gerada para a topologia USA. Sob altas cargas,

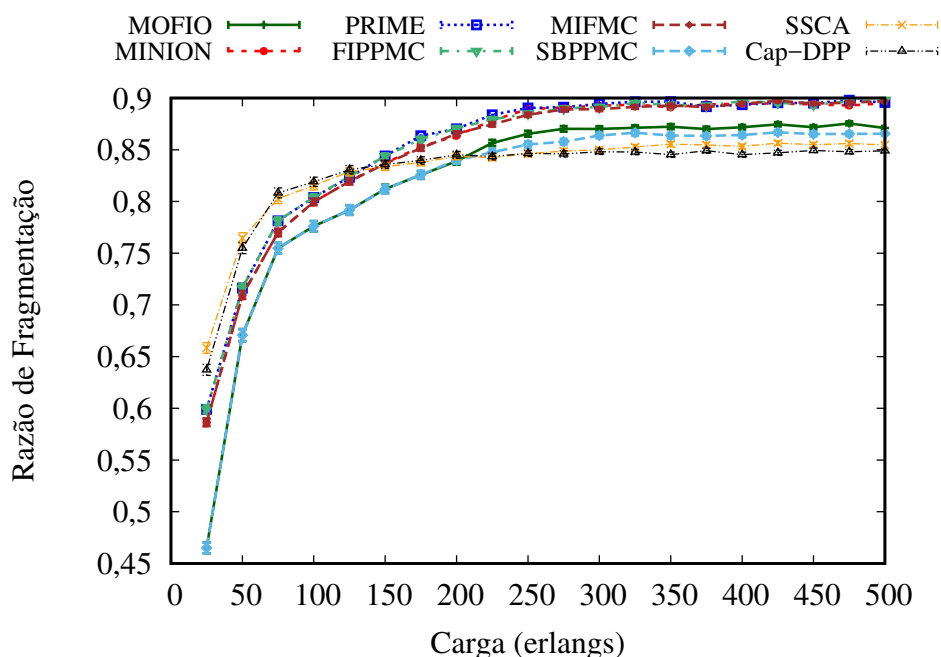


Figura 7.9: Razão de fragmentação dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.

o algoritmo PRIME seguido dos algoritmos FIPPMC, MINION e MIFMC, respectivamente, produzem as maiores relações de fragmentação, pois os algoritmos que utilizam *p-cycle* FIPP utilizam mais recursos de backup, apesar dos recursos não serem utilizados, a reserva gera fragmentação. Sob baixas cargas, os algoritmos MOFIO e SBPPMC produzem as menores relações de fragmentação, pois estes algoritmos encontram caminhos com menores números de saltos. Sob baixas cargas, os algoritmos CaPP-DPP e SSCA produzem a maior razão de fragmentação, isto acontece devido a forma que estes algoritmos utilizam para encontrar os caminhos. Conforme estes algoritmos diminuem o número de requisições aceitas, o número de desconexões juntamente com a razão de fragmentação decrementam. O algoritmo MOFIO produz sempre menor razão de fragmentação que os algoritmos MINION e PRIME, que utilizam *p-cycle* FIPP. Isto ocorre pois o número de saltos dos caminhos gerados pelos algoritmos que utilizam *p-cycle* produzem grande impacto na razão de fragmentação.

A Figura 7.10 mostra a eficiência energética gerada para a topologia USA. Sob cargas altas, algoritmo CaP-DPP produz a maior eficiência energética devido ao alto número de bloqueio de requisições, que faz com que apenas caminhos menores sejam estabelecidos. Conforme ocorre a necessidade de utilização de roteamento multicaminho pelos algoritmos MOFIO, MINION e PRIME, ocorre uma maior eficiência energética quando comparado aos algoritmos SBPPMC, MIFMC e FIPPMC. Os algoritmos MOFIO, MINION e PRIME produzem resultados próximos. Entre cargas de 150 a 275 erlangs, estes algoritmos decrementam a eficiência energética conforme a carga aumenta. Conforme aumenta o número de requisições bloqueadas, a eficiência energética volta a aumentar. O algoritmo SSCA que inicia o bloqueio de requisições sob baixas cargas produz eficiência energética sempre em crescimento.

A Figura 7.11 mostra a média do número médio de saltos dos caminhos primários por conexão gerada para a topologia USA. Sob cargas altas, os algoritmos MOFIO, PRIME e MINION

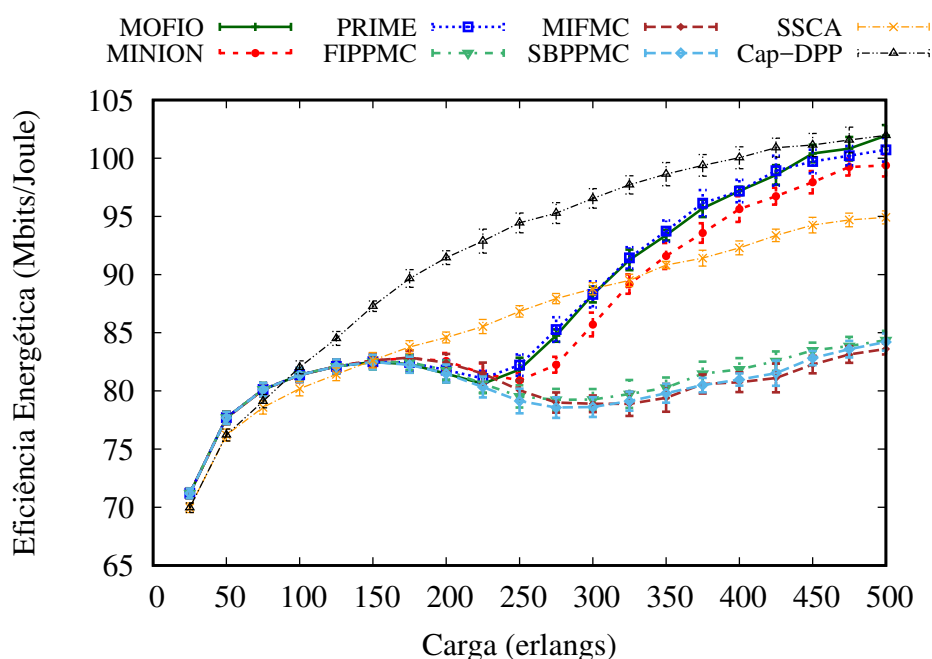


Figura 7.10: Eficiência Energética dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia USA.

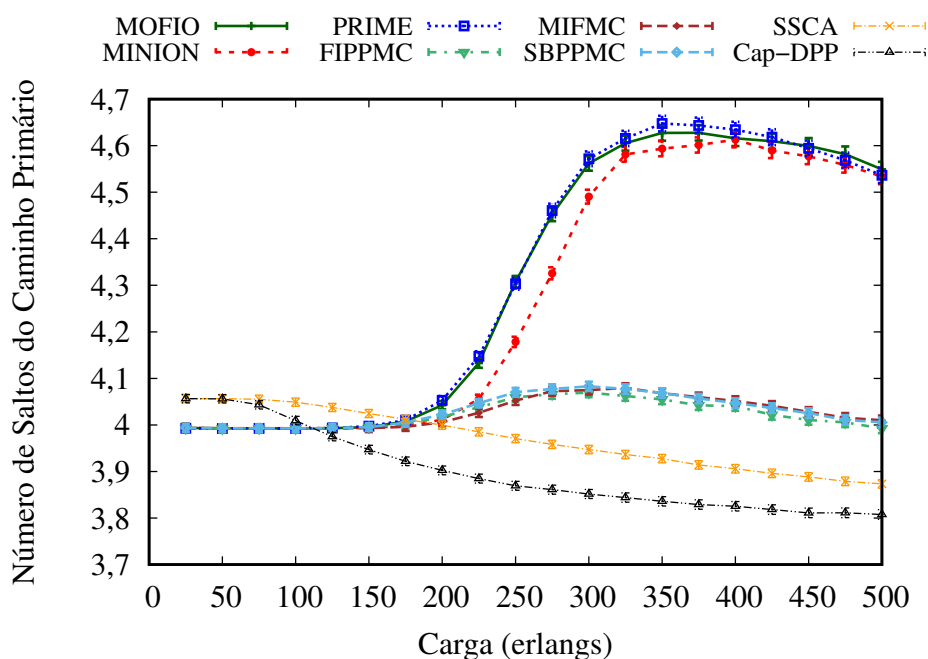


Figura 7.11: Número médio de saltos do caminho primário por conexão dos algoritmos MOFIO, MINION e PRIME em função da carga da rede para topologia USA.

que utilizam multicaminhos apresentam caminhos primários com aumento no número de saltos quando comparados com os algoritmos SBPPMC, FIPPMC e MIFMC. Para cargas baixas, em que a utilização de caminhos simples é suficiente, o número médio de saltos dos caminhos primários dos algoritmos MOFIO, PRIME e MINION são iguais aos SBPPMC, FIPPMC e MIFMC pois não utilizam multicaminhos. Ainda sob cargas baixas, o número médio de sal-

tos dos caminhos primários dos algoritmos MOFIO, PRIME e MINION são menores que os gerados para os algoritmos CaP-DPP e SSCA, pois os algoritmos MOFIO, PRIME e MINION utilizam o multigrafo, gerando uma melhor alocação e conseqüentemente caminhos mais curtos. Conforme a carga aumenta, os algoritmos CaP-DPP e SSCA aumentam o bloqueio e conseqüentemente diminuem o número de saltos. Por outro lado os algoritmos PRIME, MINION e MOFIO iniciam a utilização de roteamento multicaminho aumentando o número médio de saltos por conexão e mantendo a rejeição de requisições baixa.

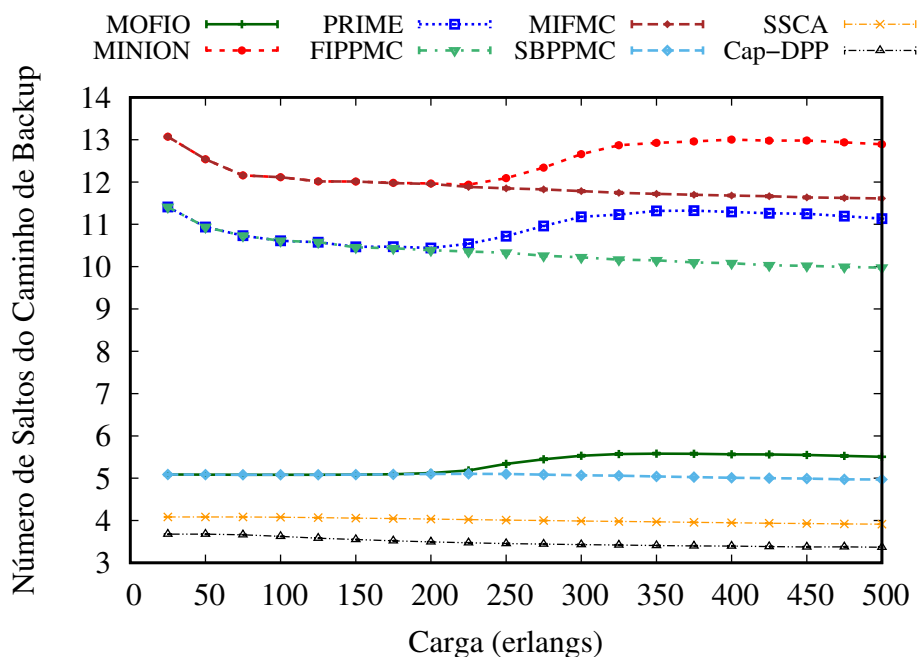


Figura 7.12: Número médio de saltos do caminho de *backup* por conexão dos algoritmos MOFIO, MINION e PRIME em função da carga da rede para topologia USA.

A Figura 7.12 mostra a média do número médio de saltos do caminho de *backup* por conexão gerado para a topologia USA. O comportamento dos saltos do caminho de *backup* para topologia USA são similares aos da topologia Pan-European. Os algoritmos que utilizam *p-cycle* produzem caminhos de *backup* com uma maior número de saltos. Sob altas cargas, os algoritmos que utilizam roteamento híbrido dividem a banda requerida por 2 caminhos, aumentando o número de saltos total quando comparado com os respectivos algoritmos que utilizam roteamento monocaminho. A diferença entre o número médio de saltos dos caminhos de *backup* dos algoritmos PRIME e MINION é maior que para topologia Pan-European, isto ocorre pois a topologia USA permite uma maior facilidade de encontrar *p-cycles* com interferência mínima. Por outro lado, o algoritmo MOFIO produz número de saltos do caminho de *backup* mais próximos que o algoritmo SSCA, ainda como consequência da conectividade da topologia.

A Figura 7.13 mostra a probabilidade de bloqueio para a topologia NSF. O algoritmo CaP-DPP e SSCA iniciaram o bloqueio de requisições sob cargas de 25 e 50 erlangs, respectivamente. O algoritmo FIPPMC e MIFMC iniciaram o bloqueio de requisições apenas sob carga de 125 erlangs, enquanto que o algoritmo SBPPMC iniciou o bloqueio de requisições apenas sob carga de 150 erlangs. O algoritmo PRIME iniciou o bloqueio de requisições apenas sob cargas de 150 erlangs. Os algoritmos MOFIO e MINION iniciaram o bloqueio de requisições

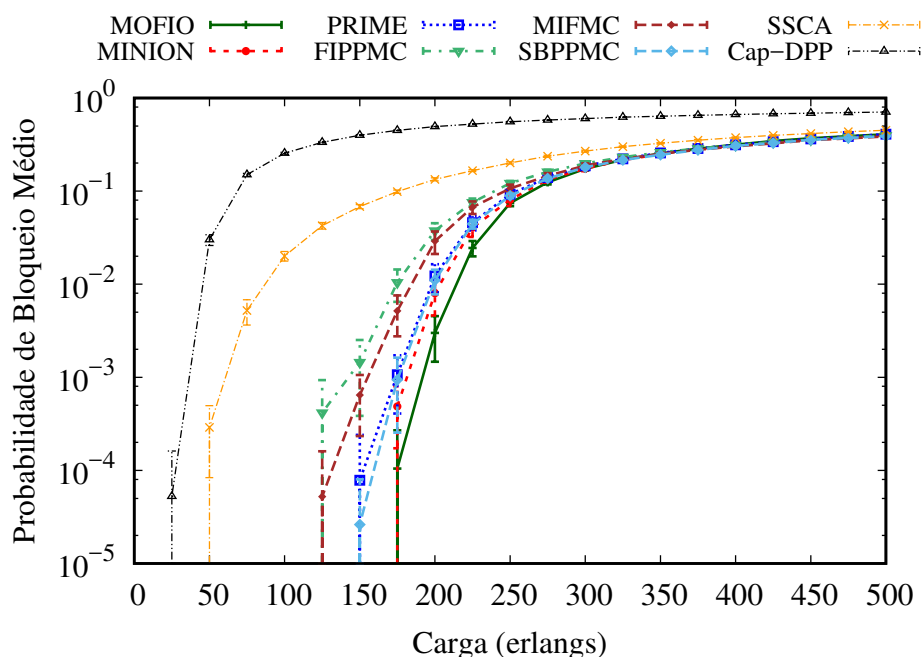


Figura 7.13: Bloqueio de banda dos algoritmos MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.

apenas sob carga de 175 erlangs. Os algoritmos que utilizam roteamento híbrido produzem menores valores de BBR que os algoritmos que utilizam roteamento simples. Sob carga de 150 erlangs, o algoritmo SBPPMC, que utiliza roteamento monocaminho, produz menores valores de BBR que o algoritmo PRIME, isto ocorre devido a menor conectividade da topologia NSF. A baixa conectividade da topologia NSF impacta nos algoritmos PRIME e MINION, que utilizam *p-cycle* e roteamento multicaminho, fazendo com que estes produzam uma maior rejeição de requisições que o algoritmo MOFIO que também utiliza roteamento multicaminho, como consequência da maior dificuldade de formar ciclos. Mesmo em uma topologia com baixa conectividade o algoritmo MINION produziu menores valores de BBR que o algoritmo PRIME, evidenciando a vantagem de utilizar interferência mínima e *p-cycle* juntamente com roteamento multicaminho. Como para as outras topologias, o algoritmo CaP-DPP produz os maiores valores de BBR, como consequência de não utilizar compartilhamento de caminhos de *backup* e não realizar boas escolhas para encontrar o caminho. O algoritmo SSCA produz o segundo maior número de requisições rejeitadas, influenciado pela não utilização de roteamento multicaminho e pela forma como o algoritmo escolhe os caminhos.

A Figura 7.14 mostra a relação de *crossstalk* por *slot* gerado para a topologia NSF. Os algoritmos PRIME e MINION iniciam os valores de CpS em 0,02 e incrementam até 0,6. Os algoritmos FIPPMC e MIFMC iniciam os valores de CpS em 0,02 e incrementam até 0,59. Os algoritmos SSCA e MOFIO iniciam os valores de CpS em 0,02 e aumentam até 0,42 e 0,51, respectivamente. O algoritmo CaP-DPP produz CpS quase constante para todas as cargas simuladas, pois este algoritmo inicia o bloqueio de requisições sob cargas baixas. Como para as outras topologias, os algoritmos FIPPMC, MIFMC, PRIME e MINION que utilizam *p-cycle* produzem os maiores valores de CpS, mais uma vez mostrando que com a maior utilização de recursos para caminhos de *backup*, maior a influência do CpS para os caminhos primários. No

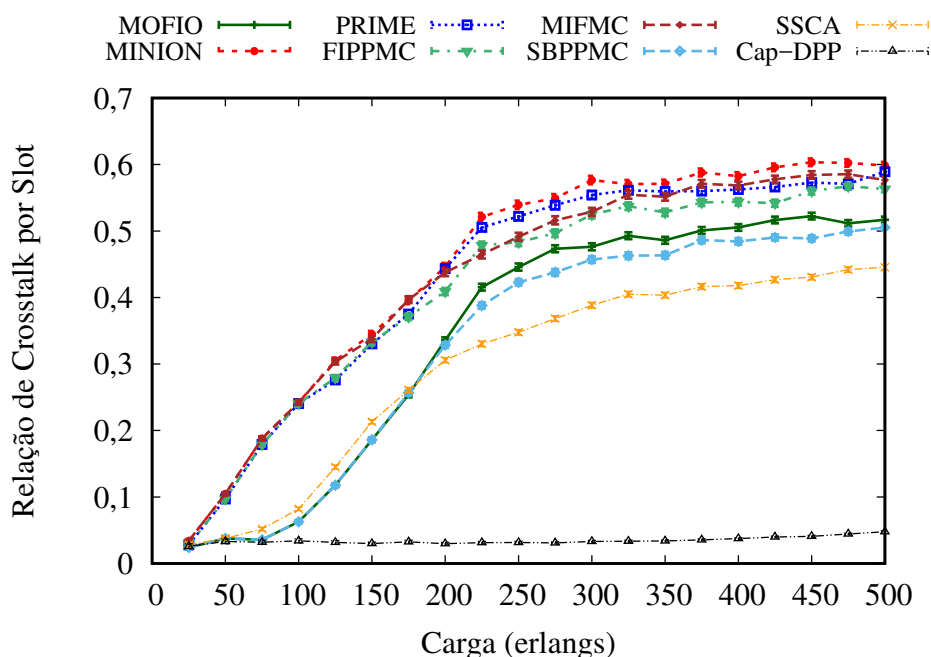


Figura 7.14: *Crosstalk* por *slot* dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.

entanto, o *crosstalk* desses algoritmos ainda estão em um limiar aceitável. Os algoritmos que utilizam roteamento híbrido produzem valores de CpS próximos ao respectivos algoritmos que utilizam roteamento monocaminho.

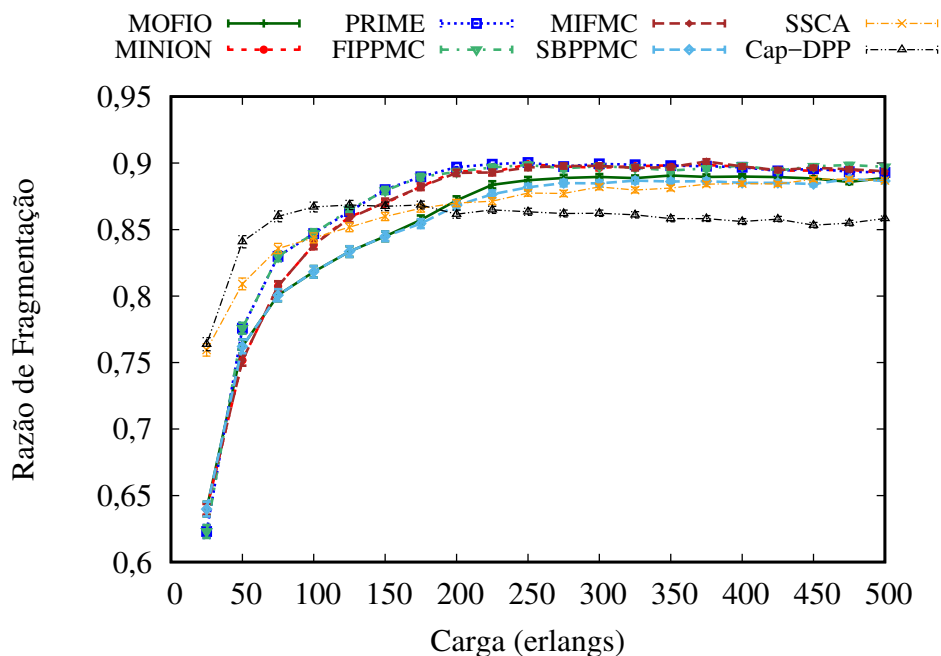


Figura 7.15: Razão de fragmentação dos algoritmos dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.

A Figura 7.15 mostra a razão de fragmentação gerada para a topologia NSF. Sob cargas altas

os algoritmos PRIME, FIPPMC, MIFMC e MINION que utilizam p -cycle FIPP produzem as maiores razões de fragmentação, devido ao número de saltos dos caminhos produzidos o que gera um maior estabelecimento de conexões. Sob cargas altas o algoritmo CaP-DPP produz a menor razão de fragmentação, como consequência do maior bloqueio de requisições, implicando em um menor número de estabelecimentos de conexões. O algoritmo MOFIO, produziu o menor bloqueio de requisições, no entanto este algoritmo produziu menor razão de fragmentação que os algoritmos PRIME E MINION, como consequência do baixo desempenhos de algoritmos que utilizam p -cycle em topologias com baixa conectividade.

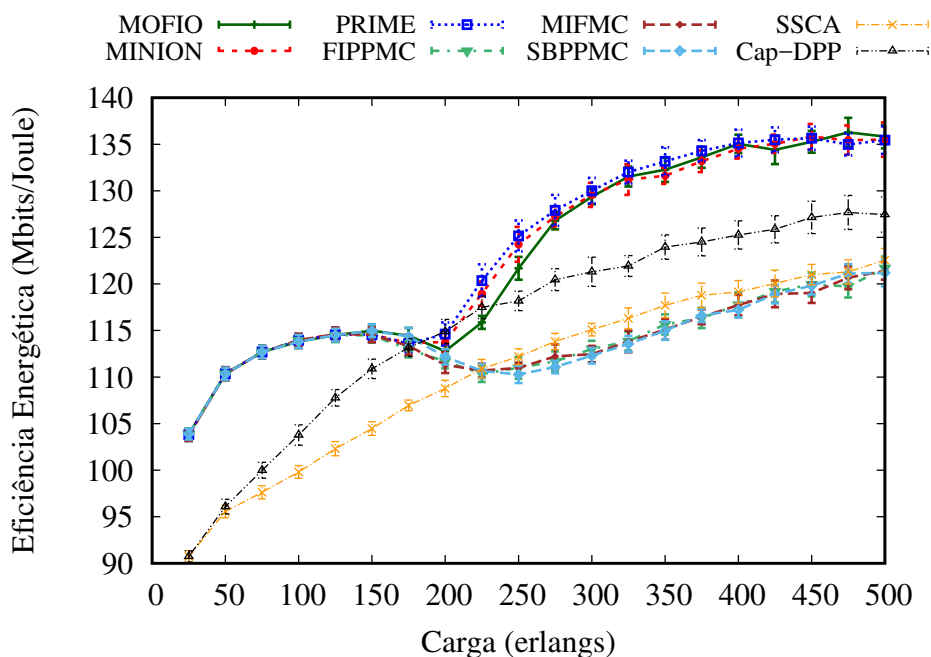


Figura 7.16: Eficiência Energética dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.

A Figura 7.16 mostra a eficiência energética gerada para a topologia NSF. Os algoritmos que utilizam roteamento multicaminho produzem a maior eficiência energética, isto acontece devido ao maior balanceamento na utilização de enlaces oferecido pelo roteamento multicaminho. Além disso, sob cargas baixas, o algoritmo SSCA produz a pior eficiência energética, pois este algoritmo produz caminhos maiores. A diferença nas curvas que representam os algoritmos MOFIO, PRIME e MINION com relação aos algoritmos SBPPMC, FIPPMC e MIFMC, respectivamente, sob cargas maiores que 200 erlangs é consequência da utilização de roteamento multicaminho, aumentando o número médio de saltos do caminho primário e o uso dos recursos da rede.

A Figura 7.17 mostra o número médio de saltos do caminho primário por conexão gerado para a topologia NSF. Como para as outras topologias, conforme a carga aumenta os algoritmos SBPPMC, FIPPMC e MIFMC CaP-DPP e SSCA, que não utilizam roteamento multicaminho, aumentam o bloqueio de requisições e decrementam o número de saltos dos caminhos primários gerados. Por outro lado, sob cargas maiores que 150 erlangs, os algoritmos MOFIO, PRIME e MINION, que utilizam roteamento multicaminho, aumentam o número de saltos total dos caminhos primários. Sob cargas altas, a rede sobrecarrega e os caminhos simples não podem ser

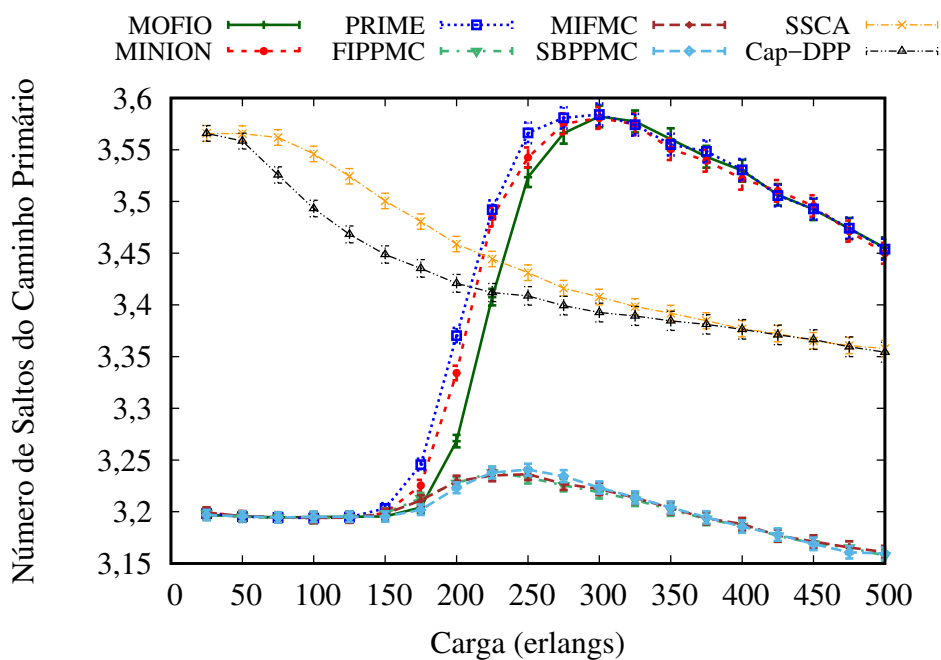


Figura 7.17: Número médio de saltos do caminho primário dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.

estabelecidos, necessitando da utilização de roteamento multicaminho. A partir de 300 erlangs, o bloqueio de requisições aumenta e o número de saltos dos caminhos produzidos diminuem.

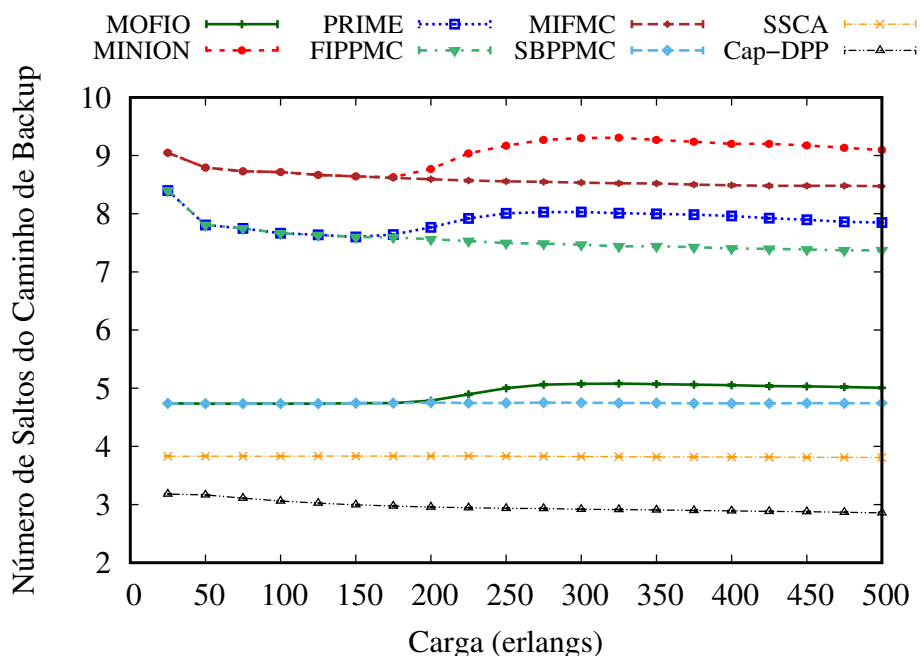


Figura 7.18: Número médio de saltos do caminho de *backup* por requisição dos algoritmos dos algoritmos SBPPMC, FIPPMC, MIFMC, SSCA, Cap-DPP, MOFIO, MINION e PRIME em função da carga da rede para topologia NSF.

A Figura 7.18 mostra a média do número médio de saltos do caminho de *backup* por conexão

gerado para a topologia NSF. O comportamento dos saltos do caminho de *backup* para topologia NSF são similares aos das outras topologias. Os algoritmos que utilizam *p-cycle* produzem caminhos com maior número médio de saltos do que os caminhos de *backup* normais. Sob altas cargas, os algoritmos que utilizam roteamento híbrido dividem a banda requerida por 2 caminhos, aumentando o número de saltos total quando comparado com os respectivos algoritmos que utilizam roteamento monocaminho. Como para as outras topologias os algoritmos PRIME e MINION, que utilizam *p-cycle*, utilizam caminhos de *backup* com maior número médio de saltos, no entanto este número não muda muito quando utilizado roteamento multicaminho, evidenciando um maior compartilhamento para proteção.

7.7 Síntese do capítulo

Neste capítulo, foram introduzidos os algoritmos MINION, PRIME e MOFIO para proteção de redes ópticas elásticas com multiplexação espacial utilizando multicaminho. Além de introduzir os algoritmos, o capítulo apresenta uma comparação dos algoritmos que utilizam roteamento multicaminho com os correspondentes algoritmos que empregam caminhos simples, bem como outros da literatura.

Os resultados comprovam que o uso de multicaminhos implica em um maior uso de banda de guarda, já que caminhos diferentes necessitam ser separados por banda de guarda. Fazendo com que haja maior consumo de recursos na rede, no entanto, isso não produz maior bloqueio de requisições, já que a divisão de caminhos utilizando roteamento multicaminho possibilita uma maior facilidade para respeitar as restrições de continuidade e contiguidade de núcleo e espectro imposta por redes ópticas elásticas com multiplexação espacial.

Os resultados mostraram que algoritmos que utilizam roteamento multicaminho conseguem prover uma maior eficiência energética, devido ao balanceamento de recursos utilizados na rede.

A utilização de multicaminhos sofre grande influencia da conectividade da topologia, isto ocorre pois topologias com baixa conectividade dificultam encontrar caminhos diferentes, impossibilitando o uso de multicaminho.

Capítulo 8

Considerações Finais e Perspectivas Futuras

8.1 Considerações Finais

Nesta Tese, realizou-se um amplo estudo sobre proteção em redes ópticas elásticas com multiplexação espacial. Foram desenvolvidos e avaliados algoritmos para proteção e estabelecimento de conexões. Os algoritmos foram avaliados para diferentes topologias e métricas. Os algoritmos propostos utilizam técnicas como: multigrafo, agregação de tráfego, sobreposição de espectro, modulação adaptativa e roteamento multicaminho.

Os algoritmos desenvolvidos objetivaram potencializar a aceitação de requisições para estabelecimento de conexões, levando em consideração o *crossstalk* gerado, a fragmentação dos enlaces da rede, a redução do consumo energético, e o número de saltos do caminho primário e de *backup*, aprimorando, assim, o uso da tecnologia de redes ópticas elásticas com multiplexação espacial.

No Capítulo 3, foram introduzidos algoritmos para proteção de caminhos ópticos. As soluções propostas avançam o estado da arte e são soluções eficientes para se prover proteção às redes ópticas elásticas com multiplexação espacial. Os resultados mostraram que a técnica do multigrafo proposta possibilita a criação de caminhos primários e de *backup* com baixo número de saltos. Os algoritmos FIPPMC, MIFMC e SBPPMC produzem bloqueio aceitável para redes ópticas elástica com multiplexação espacial, apesar da reserva de largura de banda para pré-provisionamento de caminhos de *backup*. Além disso, quanto maior a conectividade de rede, mais indicado o uso do *p-cycle* FIPP, uma vez que os algoritmos que usam o *p-cycle* FIPP produzem valores de BBR mais baixo.

No Capítulo 4, investigou-se o problema de proteção utilizando modulação adaptativa. As soluções propostas ajudaram, também, a avançar o estado da arte. Além disso, as soluções propostas aumentaram o número de requisições aceitas, diminuindo o consumo energético da rede. Os resultados mostraram uma grande redução na BBR produzida para os algoritmos PERFECTA, BARTRMAN e MMIFMC com relação às apresentadas nos algoritmos FIPPMC, MIFMC e SBPPMC. Os valores de BBR produzidos pelos algoritmos PERFECTA, BARTRMAN e MMIFMC evidenciam o benefício de escolher conjuntamente a rota, núcleo e modulação quando comparados à escolha em diferentes etapas. A utilização de diferentes formatos

de modulação melhora a utilização dos recursos disponíveis, de forma que caminhos menores suportam maior transferência de dados.

No Capítulo 5, realizou-se um estudo sobre agregação de tráfego e sobreposição de espectro para proteção de redes ópticas com multiplexação espacial. O estudo lida com proteção de caminhos que fazem sobreposição do espectro seguindo restrições pré-definidas. Além disso, os algoritmos propostos podem utilizar agregação de tráfego tanto nos caminhos primários quanto nos caminhos de *backup*. Os algoritmos propostos diminuem os recursos utilizados no caminho primário e para realizar proteção, mantendo a proteção das requisições. Os resultados mostraram uma grande redução no BBR produzida para os algoritmos SBRC, STOP, FRSCA com relação aos FIPPMC, MIFMC e SBPPMC. Apesar da agregação de tráfego prover um ganho na utilização de recursos, a sobreposição de espectro é a principal responsável pela geração de valores de menor BBR, visto que ela consegue reduzir de forma considerável os recursos utilizados para proteção. Além disto, o algoritmo SSSPF que provê proteção contra dupla falhas simultâneas foi proposto. Este algoritmo produz altos valores de BBR quando comparado com outros algoritmos. No entanto, estes resultados eram esperados, visto que as topologias reais utilizadas não permitem dupla redundância em todas as conexões da rede.

No Capítulo 6, utilizou-se o estudo dos capítulos 4 e 5 para propor novos algoritmos que utilizam a junção de todas as técnicas estudadas anteriormente. Os algoritmos RAMBO, APPROVE e MAROTO propostos produzem menor bloqueio e maior uso dos recursos da rede que os outros algoritmos propostos até o momento. Além disso, notou-se que a utilização em conjunto de *p-cycle* com as técnicas de modulação adaptativa, agregação de tráfego e sobreposição de espectro possibilitou aos algoritmos que utilizam *p-cycle* resultados comparáveis aos que utilizam caminhos de *backup* compartilhados mesmo em topologias com baixa conectividade. Os algoritmos propostos no capítulo 6 produziram os menores valores de BBR dentre todos os algoritmos propostos. Os resultados gerados demonstram a viabilidade que a utilização conjunta destas técnicas propostas podem prover.

No Capítulo 7, analisou-se o problema de proteção em redes ópticas elásticas utilizando roteamento híbrido. Os algoritmos aqui propostos priorizam a utilização de roteamento simples e caso não seja possível a utilização de caminhos simples o roteamento multicaminho é utilizado. Nos caso em que o roteamento multicaminho é utilizado, o número de caminhos necessários e a quantidade de recursos alocados para cada caminho são os dois principais parâmetros, tendo um grande impacto na eficiência espectral. Os resultados comprovaram que o uso de roteamento híbrido nos algoritmos PRIME, MINION e MOFIO aumentam o número de banda de guarda, já que caminhos diferentes necessitam ser separados por banda de guarda, fazendo com que haja maior consumo de recursos na rede. No entanto, isso não produz maior bloqueio de requisições, já que a divisão de caminhos utilizando roteamento multicaminho possibilita uma maior facilidade para respeitar as restrições de continuidade e contiguidade de núcleo e espectro imposta por redes ópticas elásticas com multiplexação espacial. Além disso, os resultados mostraram que algoritmos que utilizam roteamento multicaminho conseguem prover uma maior eficiência energética, devido ao balanceamento de recursos utilizados na rede.

Apesar dos algoritmos propostos terem sido extensivamente avaliados e terem apresentado resultados numericamente satisfatórios, este tópico de pesquisa é muito recente e existem vários problemas em aberto para investigação em trabalhos futuros.

8.2 Limitações

A avaliação dos algoritmos propostos supõe que todos os nós e enlaces da rede dispõem de características de redes utilizando multiplexação espacial. No entanto, tais características não são existentes nos atuais equipamentos de redes. Dessa forma, caso algumas características consideradas não sejam implementadas nas redes futuras, algumas modificações nos algoritmos propostos serão necessárias para que se mantenham os bons resultados alcançados através de simulações nos cenários de redes reais.

8.3 Perspectivas Futuras

A seguir são apresentadas algumas ideias para trabalhos futuros:

- Desenvolver algoritmos híbridos utilizando roteamento multicaminho em conjunto com modulação adaptativa.
- Gerar algoritmos que forneçam diferentes classes de serviços de sobrevivência para tráfego dinâmico em redes ópticas elásticas com multiplexação por divisão espacial, considerando que conexões com diferentes requisitos de confiabilidade serão classificadas por um valor de QoP, que indicará a taxa de necessidade de proteção de acordo com a classe da requisição.
- Investigar a utilização de multicaminhos de forma a escolher o número de caminhos de forma automatizada de acordo com cada requisição, não fixando o número de caminhos utilizados para cada requisição.
- Desenvolver algoritmos de agregação de tráfego que realize a postergação do momento de início da transmissão das conexões a fim de agregar lotes de conexões, com a finalidade de aumentar o número de agregações de tráfego e, conseqüentemente, reduzindo os recursos reservados.

Referências Bibliográficas

- [1] R. Asthana. *Study of p-Cycle based Protection in Optical Networks and Removal of its Shortcomings*. PhD thesis, Indian Institute Of Technology, 2007.
- [2] R. Asthana and Y. N. Singh. Protection and restoration in optical networks. *IETE Journal of Research*, 50(5):319–329, 2004.
- [3] R. Asthana, Y.N. Singh, and W.D. Grover. p-cycles: An overview. *Communications Surveys Tutorials, IEEE*, 12(1):97–111, 2010.
- [4] S. Berdagué and P. Facq. Mode division multiplexing in optical fibers. *Appl. Opt.*, 21(11):1950–1955, Jun 1982.
- [5] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing*. The Springer International Series in Engineering and Computer Science. Springer US, 1999.
- [6] F. J. Blouin, A. Sack, and W. D. Grover. Benefits of p-cycles in a mixed protection and restoration approach. In *Fourth International Workshop on Design of Reliable Communication Networks, 2003. (DRCN 2003). Proceedings.*, pages 203–210, Oct 2003.
- [7] J. J. Castro, S. Fernández, I. de Miguel, R. J. Durán, N. Fernández, N. Merayo, J. C. Aguado, P. Fernández, R. M. Lorenzo, and E. J. Abril. A comparison of dynamic traffic grooming algorithms for elastic optical networks. In *2015 17th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, July 2015.
- [8] B. C. Chatterjee, N. Sarma, and E. Oki. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Communications Surveys Tutorials*, 17(3):1776–1800, thirdquarter 2015.
- [9] X. Chen, M. Chamania, A. Jukan, A. C. Drummond, and N. L. S. d. Fonseca. On the benefits of multipath routing for distributed data-intensive applications with high bandwidth requirements and multidomain reach. In *2009 Seventh Annual Communication Networks and Services Research Conference*, pages 110–117, May 2009.
- [10] X. Chen, M. Chamania, A. Jukan, A. C. Drummond, and N. L. S. da Fonseca. QoS-constrained multi-path routing for high-end network applications. In *IEEE INFOCOM Workshops 2009*, pages 1–6, April 2009.
- [11] X. Chen, A. Jukan, A. C. Drummond, and N. L. S. da Fonseca. A multipath routing mechanism in optical networks with extremely high bandwidth requests. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pages 1–6, Nov 2009.

- [12] X. Chen, Y. Zhong, and A. Jukan. Multipath routing in elastic optical networks with distance-adaptive modulation formats. In *2013 IEEE International Conference on Communications (ICC)*, pages 3915–3920, June 2013.
- [13] X. Chen, S. Zhu, L. Jiang, and Z. Zhu. On spectrum efficient failure-independent path protection p-cycle design in elastic optical networks. *Lightwave Technology Journal of*, 33(17):3719–3729, Sept 2015.
- [14] I. Cidon, R. Rom, and Y. Shavitt. Analysis of multi-path routing. *Networking, IEEE/ACM Transactions on*, 7(6):885–896, Dec 1999.
- [15] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, and P. Demeester. Data-centric optical networks and their survivability. *IEEE Journal on Selected Areas in Communications*, 20(1):6–20, Jan 2002.
- [16] Tequila Consortium. A survey of multipath routing for traffic engineering. *Technical report, Informations and Communications University*, 2002.
- [17] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms second edition, 2001.
- [18] L. R. Costa and A. C. Drummond. New distance-adaptive modulation scheme for elastic optical networks. *IEEE Communications Letters*, PP(99):1–1, 2016.
- [19] L. R. Costa and A. C. Drummond. Novo algoritmo RMLSA com roteamento multihop em redes Ópticas elásticas. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2016*, Salvador, Bahia, may 2016.
- [20] L. R. Costa, Guilherme N. Ramos, and André C. Drummond. Leveraging adaptive modulation with multi-hop routing in elastic optical networks. *Computer Networks*, 105:124 – 137, 2016.
- [21] J. de Santi, A. C. Drummond, N. L. S. da Fonseca, X. Chen, and A. Jukan. Leveraging multipath routing and traffic grooming for an efficient load balancing in optical networks. In *2012 IEEE International Conference on Communications (ICC)*, pages 2989–2993, June 2012.
- [22] M. N. Dharmaweera, J. Zhao, L. Yan, M. Karlsson, and E. Agrell. Traffic-grooming- and multipath-routing-enabled impairment-aware elastic optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, 8(2):58–70, Feb 2016.
- [23] A. C. Drummond, N.L.S. da Fonseca, X. Chen, and A. Jukan. Roteamento multicaminho para provisão eficiente de recursos interdomínio com qualidade de serviço em redes ópticas wdm. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2017*, Belém, Pará, may 2017.
- [24] R. Essiambre, G. Kramer, P.J. Winzer, G.J. Foschini, and B. Goebel. Capacity limits of optical fiber networks. *Lightwave Technology, Journal of*, 28(4):662–701, Feb 2010.

- [25] Z. Fan, Y. Qiu, and C. Chan. Dynamic multipath routing with traffic grooming in ofdm-based elastic optical path networks. *Lightwave Technology, Journal of*, 33(1):275–281, Jan 2015.
- [26] G. B. Figueiredo, N. L. S. da Fonseca, and J. A. S. Monteiro. A minimum interference routing algorithm. In *Communications, 2004 IEEE International Conference on*, volume 4, pages 1942–1947 Vol.4, June 2004.
- [27] G. B. Figueiredo, Nelson L.S. da Fonseca, and José A.S. Monteiro. A minimum interference routing algorithm with reduced computational complexity. *Computer Networks*, 50(11):1710 – 1732, 2006.
- [28] Alexandre Fontinele, Iallen Santos, Juarez Nolêto Neto, Divanilson R. Campelo, and André Soares. An efficient ia-rmlsa algorithm for transparent elastic optical networks. *Computer Networks*, 118:1 – 14, 2017.
- [29] S. Fujii, Y. Hirota, H. Tode, and K. Murakami. On-demand spectrum and core allocation for multi-core fibers in elastic optical network. In *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), 2013*, pages 1–3, March 2013.
- [30] S. Fujii, Y. Hirota, H. Tode, and K. Murakami. On-demand spectrum and core allocation for reducing crosstalk in multicore fibers in elastic optical networks. *Optical Communications and Networking, IEEE/OSA Journal of*, 6(12):1059–1071, Dec 2014.
- [31] Tao Gao, Shanguo Huang, Bingli Guo, Xin Li, Qian Kong, Yu Zhou, Wenzhe Li, and Wanyi Gu. Survivable multipath provisioning with content connectivity in elastic optical datacenter networks. In *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, pages 1–3, July 2016.
- [32] D. Garrido, A. Leiva, A. Beghelli, R. Ahumada, and R. Olivares. Routing, code, modulation level and spectrum assignment (rcmlsa) algorithm for elastic optical networks. In *2016 18th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, July 2016.
- [33] R. Goscién, K. Walkowiak, M. Klinkowski, and J. Rak. Protection in elastic optical networks. *Network, IEEE*, 29(6):88–96, Nov 2015.
- [34] R. Goścień, K. Walkowiak, and M. Tornatore. Survivable multipath routing of anycast and unicast traffic in elastic optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, 8(6):343–355, June 2016.
- [35] Jonathan L Gross and Jay Yellen. *Graph theory and its applications*. Chapman and Hall/CRC, 2005.
- [36] W. D. Grover. *Mesh-based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2003.

- [37] L. Grüner-Nielsen, Y. Sun, J. W. Nicholson, D. Jakobsen, K. G. Jespersen, Jr. R. Lingle, and B. Pálsdóttir. Few mode transmission fiber with low dgd, low mode coupling, and low loss. *J. Lightwave Technol.*, 30(23):3693–3698, Dec 2012.
- [38] H. Guo, Y. Li, L. Li, and G. Shen. Adaptive modulation and regeneration-aware routing and spectrum assignment in sbpp-based elastic optical networks. *IEEE Photonics Journal*, 9(2):1–15, April 2017.
- [39] Frank Harary. Graph theory. Technical report, MICHIGAN UNIV ANN ARBOR DEPT OF MATHEMATICS, 1969.
- [40] Nora Hartsfield and Gerhard Ringel. *Pearls in graph theory: a comprehensive introduction*. Courier Corporation, 2013.
- [41] Tetsuya Hayashi, Toshiki Taru, Osamu Shimakawa, Takashi Sasaki, and Eisuke Sasaoka. Design and fabrication of ultra-low crosstalk and low-loss multi-core fiber. *Opt. Express*, 19(17):16576–16592, Aug 2011.
- [42] Tetsuya Hayashi, Toshiki Taru, Osamu Shimakawa, Takashi Sasaki, and Eisuke Sasaoka. Uncoupled multi-core fiber enhancing signal-to-noise ratio. *Opt. Express*, 20(26):B94–B103, Dec 2012.
- [43] K. Hinton, J. Baliga, R. Ayre, and R. S. Tucker. The future internet - an energy consumption perspective. In *2009 14th OptoElectronics and Communications Conference*, pages 1–2, July 2009.
- [44] K. Igarashi, T. Tsuritani, I. Morita, Y. Tsuchida, K. Maeda, M. Tadakuma, T. Saito, K. Watanabe, K. Imamura, R. Sugizaki, and M. Suzuki. 1.03-exabitskm super-nyquist-wdm transmission over 7,326-km seven-core fiber. In *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, pages 1–3, Sept 2013.
- [45] Toshihar Ito, Emmanuel Le Taillandier de Gabory, Manabu Arikawa, Yoichi Hashimoto, and Kiyoshi Fukuchi. Reduction of influence of inter-core cross-talk in mcf with bidirectional assignment between neighboring cores. In *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*, page OTh3K.2. Optical Society of America, 2013.
- [46] ITU. G.651.1: Characteristics of a 50/125 Mm multimode graded index optical fibre cable for the optical access network. March 2007.
- [47] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. *IEEE Communications Magazine*, 48(8):138–145, August 2010.
- [48] M. Jinno, H. Takara, B. Kozicki, Yukio Tsukishima, Y. Sone, and S. Matsuoka. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *Communications Magazine, IEEE*, 47(11):66–73, 2009.

- [49] N. Kadu, S. Shakya, and X. Cao. Modulation-aware multipath routing and spectrum allocation in elastic optical networks. In *2014 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, Dec 2014.
- [50] D. Klonidis, P. Zakyntinos, and I. Tomkos. Opportunities and challenges in the network planning of spatially and spectrally elastic optical networks. In *Transparent Optical Networks (ICTON), 2015 17th International Conference on*, pages 1–4, July 2015.
- [51] A. Kodian and W.D. Grover. Failure-independent path-protecting pycles: efficient and simple fully preconnected optimal-path protection. *IEEE, J. Lightwave Technol.*, 23:3241–3259, Oct. 2005.
- [52] Masanori Koshiha, Kunimasa Saitoh, Katsuhiro Takenaga, and Shoichiro Matsuo. Multi-core fiber design and analysis: coupled-mode theory and coupled-power theory. *Opt. Express*, 19(26):B102–B111, Dec 2011.
- [53] J. Li and K. L. Yeung. Efficient path protection using bi-directional wdm transmission technology. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 4, pages 5 pp.–1914, Dec 2005.
- [54] M. Liu, M. Tornatore, and B. Mukherjee. Survivable traffic grooming in elastic optical networks -shared protection. *Lightwave Technology, Journal of*, 31(6):6230–6234, March 2013.
- [55] M. Liu, M. Tornatore, and B. Mukherjee. Survivable traffic grooming in elastic optical networks shared protection. *IEEE, Journal of Lightwave Technology*, 31(6):903–909, 2013.
- [56] T. Mori, T. Sakamoto, M. Wada, T. Yamamoto, and F. Yamamoto. Six-lp-mode transmission fiber with dmd of less than 70 ps/km over c+l band. In *OFC 2014*, pages 1–3, March 2014.
- [57] P. M. Moura, N. L. S. d. Fonseca, and R. A. Scaraficci. Fragmentation aware routing and spectrum assignment algorithm. In *2014 IEEE International Conference on Communications (ICC)*, pages 1137–1142, June 2014.
- [58] P. M. Moura and N. L. S. da Fonseca. Routing, core and spectrum assignment based on connected component labelling for sdm optical networks. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2016.
- [59] P. M. Moura and A. C. Drummond. FlexGridSim: Flexible Grid Optical Network Simulator. <http://www.lrc.ic.unicamp.br/FlexGridSim/>.
- [60] P. M. Moura, R. A. Scaraficci, and N. L. S. d. Fonseca. Algorithm for energy efficient routing, modulation and spectrum assignment. In *2015 IEEE International Conference on Communications (ICC)*, pages 5961–5966, June 2015.

- [61] A. Muhammad, G. Zervas, G. Saridis, E.H. Salas, D. Simeonidou, and R. Forchheimer. Flexible and synthetic sdm networks with multi-core-fibers implemented by programmable roadms. In *Optical Communication (ECOC), 2014 European Conference on*, pages 1–3, Sept 2014.
- [62] A. Muhammad, G. Zervas, D. Simeonidou, and R. Forchheimer. Routing, spectrum and core allocation in flexgrid sdm networks with multi-core fibers. In *Optical Network Design and Modeling, 2014 International Conference on*, pages 192–197, May 2014.
- [63] B. Mukherjee. Wdm optical communication networks: progress and challenges. *Selected Areas in Communications, IEEE Journal on*, 18(10):1810–1824, 2000.
- [64] H. M. N. S. Oliveira and N. L. S. da Fonseca. Traffic grooming and spectrum overlap in fipp p-cycle for protection of elastic optical networks. In *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, Nov 2016.
- [65] H. M. N. S. Oliveira and N. L. S. da Fonseca. Algorithm for shared path for protection of space division multiplexing elastic optical networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.
- [66] H. M. N. S. Oliveira and N. L. S. da Fonseca. Protection, routing, modulation, core and spectrum allocation in sdm elastic optical networks. *IEEE Communications Letters*, pages 1–1, 2018.
- [67] H. M. N. S. Oliveira and N. L. S. da Fonseca. Spectrum overlap and traffic grooming in p-cycle algorithm protected sdm optical networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2018.
- [68] H.M.N.S. Oliveira and N.L.S. da Fonseca. Algorithm for FIPP p-cycle path protection in flexgrid networks. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 1278–1283, Dec 2014.
- [69] H.M.N.S. Oliveira and N.L.S. da Fonseca. Protection in elastic optical networks against up to two failures based FIPP p-cycle. In *Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on*, pages 369–375, May 2014.
- [70] H.M.N.S. Oliveira and N.L.S. da Fonseca. Algorithm for protection of space division multiplexing elastic optical networks. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2016.
- [71] H.M.N.S. Oliveira and N.L.S. da Fonseca. The minimum interference p-cycle algorithm for protection of space division multiplexing elastic optical networks. In *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, Nov 2016.
- [72] H.M.N.S. Oliveira and N.L.S. da Fonseca. Proteção de redes Ópticas elásticas baseada em agregação de tráfego, sobreposição de espectro e p-cycle fipp. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2016*, Salvador , Bahia, may 2016.

- [73] H.M.N.S. Oliveira and N.L.S. da Fonseca. Algoritmo de p-cycle e interferência mínima para proteção de redes Ópticas elásticas com multiplexação espacial. *IEEE Latin America Transactions*, 15(7):779–785, July 2017.
- [74] H.M.N.S. Oliveira and N.L.S. da Fonseca. Proteção de redes ópticas elásticas com multiplexação espacial baseada em modulação, p-cycle FIPP e interferência mínima. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2017*, Belém, Pará, may 2017.
- [75] H.M.N.S. Oliveira and N.L.S. da Fonseca. Routing, spectrum, core and modulation level assignment algorithm for protected sdm optical networks. In *2017 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2017.
- [76] H.M.N.S. Oliveira and N.L.S. da Fonseca. Sharing spectrum and straddling p-cycle fipp for protection against two simultaneous failures in sdm elastic optical networks. In *2017 9th IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, Nov 2017.
- [77] H.M.N.S. Oliveira and N.L.S. da Fonseca. Proteção de redes Ópticas elásticas com multiplexação espacial baseada em agregação de tráfego, sobreposição de espectro e compartilhamento de caminho de backup. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2018*, Campos do Jordão, São Paulo, may 2018.
- [78] C. Ou, J. Zhang, H. Zang, L. H. Sahasrabudde, and B. Mukherjee. New and improved approaches for shared-path protection in wdm mesh networks. *Journal of Lightwave Technology*, 22(5):1223–1232, May 2004.
- [79] L. Padhi, S. Kartikeya, K.M. Sivalingam, and S.S. Sai. Multi-path routing in optical wdm networks: Even versus uneven split bandwidth allocation. In *Signal Processing and Communications (SPCOM), 2010 International Conference on*, pages 1–5, July 2010.
- [80] R. Proietti, Lei Liu, R.P. Scott, Binbin Guan, Chuan Qin, Tiehui Su, F. Giannone, and S.J.B. Yoo. 3d elastic optical networking in the temporal, spectral, and spatial domains. *Communications Magazine, IEEE*, 53(2):79–87, Feb 2015.
- [81] S. Rai, O. Deshpande, C. Ou, C. U. Martel, and B. Mukherjee. Reliable multipath provisioning for high-capacity backbone mesh networks. *IEEE/ACM Transactions on Networking*, 15(4):803–812, Aug 2007.
- [82] Rajiv Ramaswami and Kumar N. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition edition, 2002.
- [83] D. J. Richardson, J. M. Fini, and L. E. Nelson. Space-division multiplexing in optical fibres. *Nat. Photonics*, vol. 7, no. 5, pp. 354–362, 2013.
- [84] L. Ruan and Y. Zheng. Dynamic survivable multipath routing and spectrum allocation in ofdm-based flexible optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, 6(1):77–85, Jan 2014.

- [85] R. Ryf, R. J. Essiambre, S. Randel, M. A. Mestre, C. Schmidt, and P. J. Winzer. Impulse response analysis of coupled-core 3-core fibers. In *2012 38th European Conference and Exhibition on Optical Communications*, pages 1–3, Sept 2012.
- [86] Jun Sakaguchi, Benjamin J. Puttnam, Werner Klaus, Yoshinari Awaji, Naoya Wada, Atsushi Kanno, Tetsuya Kawanishi, Katsunori Imamura, Harumi Inaba, Kazunori Mukasa, Ryuichi Sugizaki, Tetsuya Kobayashi, and Masayuki Watanabe. 305 tb/s space division multiplexed transmission using homogeneous 19-core fiber. *J. Lightwave Technol.*, 31(4):554–562, Feb 2013.
- [87] N. Sambo, F. Cugini, G. Bottari, P. Iovanna, and P. Castoldi. Distributed setup in optical networks with flexible grid. In *2011 37th European Conference and Exhibition on Optical Communication*, pages 1–3, Sept 2011.
- [88] N. Sambo, G. Meloni, F. Cugini, A. D’Errico, L. Potì, P. Iovanna, and P. Castoldi. Routing, code, and spectrum assignment (rcsa) in elastic optical networks. In *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, March 2015.
- [89] N. Sambo, G. Meloni, F. Cugini, A. D’Errico, L. Potì, P. Iovanna, and P. Castoldi. Routing code and spectrum assignment (rcsa) in elastic optical networks. *Journal of Lightwave Technology*, 33(24):5114–5121, Dec 2015.
- [90] N. Sambo, G. Meloni, F. Cugini, F. Fresi, A. D’Errico, L. Potì, P. Iovanna, and P. Castoldi. Routing, code, and spectrum assignment, subcarrier spacing, and filter configuration in elastic optical networks [invited]. *IEEE/OSA Journal of Optical Communications and Networking*, 7(11):B93–B100, November 2015.
- [91] G. M. Saridis, D. Alexandropoulos, G. Zervas, and D. Simeonidou. Survey and evaluation of space division multiplexing: From technologies to optical networks. *IEEE Communications Surveys Tutorials*, 17(4):2136–2156, Fourthquarter 2015.
- [92] Pouria Sayyad Khodashenas, Jaume Comellas, Salvatore Spadaro, and Jordi Perelló. Dynamic source aggregation of subwavelength connections in elastic optical networks. *Photonic Network Communications*, 26(2):131–139, Dec 2013.
- [93] X. Shao, Y. Yeo, Z. Xu, X. Cheng, and L. Zhou. Shared-path protection in ofdm-based optical networks with elastic bandwidth allocation. In *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference*, pages 1–3, March 2012.
- [94] B. Shariati, P. S. Khodashenas, J. M. Rivas-Moscoso, S. Ben-Ezra, D. Klonidis, F. Jiménez, L. Velasco, and I. Tomkos. Evaluation of the impact of different sdm switching strategies in a network planning scenario. In *2016 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, March 2016.
- [95] P. Sillard, D. Molin, M. Bigot-Astruc, H. Maerten, D. Van Ras, and F. Achten. Low-dm-gd 6-lp-mode fiber. In *OFC 2014*, pages 1–3, March 2014.

- [96] S. K. Singh, T. Das, and A. Jukan. A survey on internet multipath routing and provisioning. *IEEE Communications Surveys Tutorials*, 17(4):2157–2175, Fourthquarter 2015.
- [97] H. Takara, A. Sano, T. Kobayashi, H. Kubota, H. Kawakami, A. Matsuura, Y. Miyamoto, Y. Abe, H. Ono, K. Shikama, Y. Goto, K. Tsujikawa, Y. Sasaki, I. Ishida, K. Takenaga, S. Matsuo, K. Saitoh, M. Koshiba, and T. Morioka. 1.01-pb/s (12 sdm/222 wdm/456 gb/s) crosstalk-managed transmission with 91.4-b/s/hz aggregate spectral efficiency. In *European Conference and Exhibition on Optical Communication*, page Th.3.C.1. Optical Society of America, 2012.
- [98] Katsuhiko Takenaga, Yoko Arakawa, Yusuke Sasaki, Shoji Tanigawa, Shoichiro Matsuo, Kunimasa Saitoh, and Masanori Koshiba. A large effective area multi-core fiber with an optimized cladding thickness. *Opt. Express*, 19(26):B543–B550, Dec 2011.
- [99] Y. Takita, K. Tajima, T. Hashiguchi, T. Katagiri, and T. Naito. Impact of adaptive modulation on cost efficient traffic grooming in elastic optical networks. In *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, March 2015.
- [100] Yuanlong Tan, Ruijie Zhu, Hui Yang, Yongli Zhao, Jie Zhang, Zhu Liu, Qinghai Qu, and Ziguan Zhou. Crosstalk-aware provisioning strategy with dedicated path protection for elastic multi-core fiber networks. In *2016 15th International Conference on Optical Communications and Networks (ICOON)*, pages 1–3, Sept 2016.
- [101] T. Tanaka, T. Inui, and W. Imajuku. A static traffic grooming algorithm for elastic optical networks with adaptive modulation. In *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, pages 1–3, July 2016.
- [102] H. Tode and Y. Hirota. Routing, spectrum and core assignment for space division multiplexing elastic optical networks. In *Telecommunications Network Strategy and Planning Symposium (Networks), 2014 16th International*, pages 1–7, Sept 2014.
- [103] H. Tode and Y. Hirota. Routing, spectrum and core assignment on sdm optical networks (invited). In *2016 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, March 2016.
- [104] Jorge López Vizcaíno, Yabin Ye, and Idelfonso Tafur Monroy. Energy efficiency analysis for flexible-grid ofdm-based optical networks. *Computer Networks*, 56(10):2400 – 2419, 2012. Green communication networks.
- [105] Srinivas Vutukury and J. J. Garcia-Luna-Aceves. A simple approximation to minimum-delay routing. *SIGCOMM Comput. Commun. Rev.*, 29(4):227–238, August 1999.
- [106] Peter Winzer, Alan Gnauck, Agnieszka Konczykowska, Filipe Jorge, and Jean-Yves Dupuy. Penalties from in-band crosstalk for advanced optical modulation formats. In *37th European Conference and Exposition on Optical Communications*, page Tu.5.B.7. Optical Society of America, 2011.

- [107] N. Xiao and L. Ruan. Survivable multipath provisioning in ofdm-based flexible optical networks. In *2012 IEEE Globecom Workshops*, pages 346–351, Dec 2012.
- [108] K. Xie, H. Tao, X. Wang, G. Xie, J. Wen, J. Cao, and Z. Qin. Divide and conquer for fast srlg disjoint routing. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 622–633, June 2018.
- [109] H. Xuan, Y. Wang, S. Hao, Z. Xu, X. Li, and X. Gao. Security-aware routing and core allocation in elastic optical network with multi-core. In *2016 12th International Conference on Computational Intelligence and Security (CIS)*, pages 294–298, Dec 2016.
- [110] X.Xie, Xin Li, S. Huang, B. Guo, S. Yin, Q. Kong, T. Gao, and W. Zhai. Design for shared backup path protection based on content connectivity against disaster in elastic optical datacenter networks. In *2016 15th International Conference on Optical Communications and Networks (ICOON)*, pages 1–3, Sept 2016.
- [111] Z. Ye, A. N. Patel, P. N. Ji, and C. Qiao. Distance-adaptive and fragmentation-aware optical traffic grooming in flexible grid optical networks. In *2014 OptoElectronics and Communication Conference and Australian Conference on Optical Fibre Technology*, pages 355–356, July 2014.
- [112] S. Yin, S. Huang, B. Guo, X. Li, C. Wang, and H. Huang. Inter-core crosstalk aware routing, spectrum and core allocation in multi-dimensional optical networks. In *2017 Opto-Electronics and Communications Conference (OECC) and Photonics Global Conference (PGC)*, pages 1–4, July 2017.
- [113] Shan Yin, Shanguo Huang, Min Zhang, Bingli Guo, Jie Zhang, and Wanyi Gu. Dynamic routing, modulation level and spectrum allocation (rmlsa) in fwdm with modulation format conversion. *Optik - International Journal for Light and Electron Optics*, 125(11):2597 – 2601, 2014.
- [114] Cunqian Yu, Weigang Hou, Ying Wu, Jingjing Wu, and Zhimin Sun. Adaptive multilevel modulation for grooming in elastic cloud optical networks. *Photonic Netw. Commun.*, 31(3):524–531, June 2016.
- [115] Guoying Zhang, M. De Leenheer, and B. Mukherjee. Optical traffic grooming in ofdm-based elastic optical networks [invited]. *Optical Communications and Networking, IEEE/OSA Journal of*, 4(11):B17–B25, Nov 2012.
- [116] J. Zhang, Y. Ji, M. Song, Y. Zhao, X. Yu, J. Zhang, and B. Mukherjee. Dynamic traffic grooming in sliceable bandwidth-variable transponder-enabled elastic optical networks. *Journal of Lightwave Technology*, 33(1):183–191, Jan 2015.
- [117] J. Zhang and B. Mukherjee. A review of fault management in wdm mesh networks: basic concepts and research challenges. *IEEE Network*, 18(2):41–48, Mar 2004.
- [118] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee. Energy efficiency in telecom optical networks. *IEEE Communications Surveys Tutorials*, 12(4):441–458, Fourth 2010.

- [119] Yongjun Zhang, Lei Yan, Haotian Wang, and Wanyi Gu. Routing, wavelength and mode assignment algorithm for space division multiplexing transmission network. In *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on*, pages 1383–1385, Dec 2012.

Apêndice A

Flexgridsim

Com o início da idealização das redes ópticas elásticas, surgiu a necessidade do desenvolvimento de um ambiente de simulação no qual os algoritmos propostos relacionados, pudessem ser implementado e avaliados. O aluno de Doutorado Pedro Mesquita Moura do laboratório de redes de computadores, utilizou a base do simulador WDMSim, simulador bastante difundido e utilizado para redes ópticas WDM, e desenvolveu o simulador Flexgridsim.

O simulador Flexgridsim é um simulador de eventos discretos para redes ópticas elásticas desenvolvido em Java, capaz de simular tráfego com chegadas e partidas de requisições de conexão com capacidade de agregação. Para cada chegada, o simulador executa o algoritmo de agregação/RSA/RMLSA avaliado, que é responsável pela aceitação ou bloqueio da chamada. Todas as operações necessárias para a obtenção de informações sobre o estado da rede, a alocação de caminhos ópticos e aceitação ou bloqueio das chamadas são acessíveis através de uma interface simples, o que facilita a adição de novos algoritmos ao simulador.

Após a fase de validação e experimentação, o simulador FlexGridSim foi disponibilizado para a comunidade através do endereço eletrônico: <http://www.lrc.ic.unicamp.br/flexgridsim/>

Em meados de 2014 o simulador recebeu atualizações (Flexgridsim 2.0), desta vez suportando simulações de redes ópticas elásticas com multiplexação espacial. Podendo agora executar algoritmos de agregação/RCSA/RMLCSA. Além disso, nesta nova versão o simulador gera automaticamente gráficos de saída que são configurados no arquivo através de um xml.

O simulador Flexgridsim na versão 2.0 recebeu adaptações para que permitisse cenários de proteção. Cenários com proteção são ativados através do xml. Nos cenários de proteção, todas as requisições devem possuir tanto caminhos de proteção quanto caminhos de backup. Além disso, caminhos de proteção podem ser compartilhados.

O restante desse apêndice está organizado da seguinte forma: a Seção A.1 apresenta uma breve discussão sobre o projeto do simulador Flexgridsim; e na Seção A.2 são dados exemplos do uso do simulador.

A.1 Projeto do simulador Flexgridsim

O simulador Flexgridsim pode ser logicamente separado em 4 partes, a configuração do ambiente de simulação, a geração da lista de eventos, a simulação em si, e a geração dos resultados.

A.1.1 Configuração do ambiente de simulação

A configuração do ambiente de simulação é realizada através de parâmetros de um arquivo XML. A tag <rsa> fornece o nome do algoritmo de agregação/RSCA que será considerado na simulação, a tag <traffic> contém todos os dados relativos ao tráfego que será gerado, e a tag <physical-topology> declara a topologia da rede, incluindo todos os nós e enlaces.

Abaixo segue exemplo de ambiente de simulação para um topologia aleatória que possui 4 nós e 40 enlaces. A topologia física é formada por 7 núcleos, cada núcleo com 320 slots. No exemplo a proteção esta ativada, logo para toda chamada será necessário um caminho primário e um de backup.

```
<?xml version="1.0" encoding="UTF-8"?>
<flexgridsim version="2.0">
  <rsa module="MyRCSA"/>
  <trace file="trace.tr"/>

  <traffic calls="10000" load="215">
    <calls holding-time="1.0" rate="1000" cos="1" weight="1"/>
    <calls holding-time="1.0" rate="4000" cos="1" weight="1"/>
    <calls holding-time="1.0" rate="10000" cos="1" weight="1"/>
  </traffic>

  <virtual-topology name="MyVirtualTopology">
  </virtual-topology>

  <physical-topology name="MyPhysicalTopology" cores="7" protection="true" slots="240" slotsBandwidth="12.5">
    <nodes>
      <node id="0"/>
      <node id="1"/>
      <node id="2"/>
      <node id="3"/>
    </nodes>
    <!-- Peso do enlace e igual a distancia em Km e o delay e igual a lms para cada 200Km -->
    <links>
      <link id="0" source="0" destination="1" delay="3.75" bandwidth="10000" weight="750" distance="750"/>
      <link id="1" source="1" destination="0" delay="3.75" bandwidth="10000" weight="750" distance="750"/>
      <link id="2" source="0" destination="2" delay="3.75" bandwidth="10000" weight="750" distance="750"/>
      <link id="3" source="2" destination="0" delay="3.75" bandwidth="10000" weight="750" distance="750"/>
      <link id="4" source="0" destination="3" delay="6.5" bandwidth="10000" weight="1300" distance="1300"/>
      <link id="5" source="3" destination="0" delay="6.5" bandwidth="10000" weight="1300" distance="1300"/>
      <link id="6" source="1" destination="2" delay="6" bandwidth="10000" weight="1200" distance="1200"/>
      <link id="7" source="2" destination="1" delay="6" bandwidth="10000" weight="1200" distance="1200"/>
    </links>
  </physical-topology>
  <!-- Graficos gerados -->
  <graphs>
    <graph name="mbr" dots-file="bf-graph-mbr-nsf.dat"/>
    <graph name="avgcrosstalk" dots-file="bf-graph-xt-nsf.dat"/>
    <graph name="avgbps" dots-file="bf-graph-avgbps-nsf.dat"/>
  </graphs>
</flexgridsim>
```

No XML acima, o tráfego configurado é composto por 10000 chamadas com taxas de transmissão uniformemente escolhidas, sendo que todas possuem um tempo médio de duração de uma unidade de tempo. A carga simulada na rede é de 215 Erlang. A capacidade de transmissão dos comprimentos de onda de um enlace podem ser declaradas em qualquer unidade.

A.1.2 Geração dos eventos da simulação

A partir das informações existentes na tag <traffic> o simulador gera os eventos de chegada e partida baseado nas informações da carga da rede, do tempo médio de duração, e da taxa de transmissão das chamadas. Um fila de prioridade é utilizada para armazenar todos os eventos criados (chegadas e partidas). A ordenação dos elementos dessa fila é feita de acordo com o momento em que os eventos devem ocorrer durante a simulação.

A.1.3 Simulação

A execução da simulação é feita através da remoção em ordem dos eventos, a partir da fila de eventos, e a passagem do evento para o módulo de controle da simulação. Este módulo então é responsável por executar o algoritmo de roteamento definido repassando a ele o evento. Por sua vez, o algoritmo de roteamento, baseado nas informações de estado da rede e no evento recebido, toma a decisão de aceitar ou bloquear uma chamada, no caso de um evento de chegada, ou simplesmente é informado de uma partida. O algoritmo possui interfaces para alocar recursos na rede e assim poder criar as condições necessárias para o estabelecimento de uma chamada.

A.1.4 Avaliação dos resultados da simulação

Durante a simulação todos os eventos podem ser registrados em um arquivo de trace. Além disso, o simulador pode gerar arquivos *.dat* que facilitam a criação de gráficos a partir de funções pré-definidas. Desta forma, permitindo a geração do resultados de forma imediata após o término de uma simulação.

Os arquivos *.dat* de saída são configurados no arquivo xml através da tag `<graphs>`. O arquivo *.dat* gerado, é compatível com o gnuplot.

A.2 Execução do simulador Flexgridsim

Por ser implementado em Java, a execução do Flexgridsim independe de plataforma, sendo necessário apenas que haja uma máquina virtual Java (JVM 7 ou superior) instalada no sistema operacional. O simulador deve ser executado a partir da linha de comando da seguinte forma:

Uso: **java -jar FlexGridSim.jar xml_file number_of_simulations [-trace] [-verbose] [carga_mínima carga_máxima passo]**

Os parâmetros obrigatório do simulador são os seguintes:

- `simulation_file`: o arquivo XML que configura o ambiente de simulação;
- `seed`: um número que define a quantidade de conjuntos de sementes escolhidas internamente pelo simulador de forma a maximizar a qualidade das sequências aleatórias utilizadas. Um número maior que um precisa ser utilizado para gerar resultados com intervalos de confiança.

Os parâmetros opcionais são os seguintes:

- `trace`: define a geração do arquivo de trace;
- `verbose`: define o modo de geração de mensagens na tela sobre os passos da simulação;
- `carga_mínima carga_máxima passo`: permite a automação de diversas execuções de uma mesma simulação para uma faixa de cargas no intervalo [`carga_mínima`, `carga_máxima`] com o valor `passo` como incremento.

A.3 Validação

Com o objetivo de validar a ferramenta de simulação foram realizados vários testes de verificação e análises de resultados em diversas topologias de rede. Foram realizadas validações através de comparações com resultados de simulação de outros trabalhos, todas modeladas e testadas com sucesso.