



UNIVERSIDADE FEDERAL DA BAHIA

DISSERTAÇÃO DE MESTRADO

**REDUÇÃO DO IMPACTO DAS RESERVAS ANTECIPADAS
SOBRE AS IMEDIATAS EM REDES DE CIRCUITO
DINÂMICO**

Diêgo Braga Monteiro de Moura

Programa de Pós-Graduação em Ciência da Computação

Salvador
04/03/2015

PGCOMP-Msc-2015

DIÊGO BRAGA MONTEIRO DE MOURA

**REDUÇÃO DO IMPACTO DAS RESERVAS ANTECIPADAS
SOBRE AS IMEDIATAS EM REDES DE CIRCUITO DINÂMICO**

Esta Dissertação de Mestrado foi apresentada ao Mestrado Multi-institucional em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: LEOBINO NASCIMENTO SAMPAIO
Co-orientador: GUSTAVO BITTENCOURT FIGUEIREDO

Salvador
04/03/2015

Ficha catalográfica.

SEU NOME EM CITACOES

REDUÇÃO DO IMPACTO DAS RESERVAS ANTECIPADAS SOBRE AS IMEDIATAS EM REDES DE CIRCUITO DINÂMICO/ Diêgo Braga Monteiro de Moura– Salvador, 04/03/2015.

56p.: il.

Orientador: LEOBINO NASCIMENTO SAMPAIO.

Co-orientador: GUSTAVO BITTENCOURT FIGUEIREDO.

Dissertação (Mestrado)– UNIVERSIDADE FEDERAL DA BAHIA, INSTITUTO DE MATEMÁTICA, 04/03/2015.

TOPICOS PARA FICHA CATALOGRAFICA.

I. NOME DO SEU ORIENTADOR EM CITACOES. II. NOME DO SEU CO-ORIENTADOR EM CITACOES.

III. UNIVERSIDADE FEDERAL DA BAHIA. INSTITUTO DE MATEMÁTICA. IV. Título.

NUMERO CDD

TERMO DE APROVAÇÃO

DIÊGO BRAGA MONTEIRO DE MOURA

REDUÇÃO DO IMPACTO DAS RESERVAS ANTECIPADAS SOBRE AS IMEDIATAS EM REDES DE CIRCUITO DINÂMICO

Esta Dissertação de Mestrado foi julgada adequada à obtenção do título de Mestre em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 04 de Março de 2015

Prof. Dr. Artur Ziviani
Laboratório Nacional de Computação Científica

Profa. Dra. Fabíola Gonçalves Pereira Greve
Universidade Federal da Bahia

Prof. Dr. Rafael Augusto de Melo
Universidade Federal da Bahia

Dedico à minha família.

AGRADECIMENTOS

Este desafio foi alcançado através da ajuda de muitas pessoas. Primeiramente, eu quero agradecer a minha avó Aroldina por esses anos todos de suporte e atenção a mim dispensado. Você é a base para que tudo isso ocorra.

Ao meu pai, por toda a paciência que teve durante esse período difícil, em que muitas vezes eu estava de mal humor. Agradecê-lo pela presença constante na cozinha, fazendo excelentes comidas e, portanto, me liberando de alguns afazeres. Chefe, o meu muito obrigado !

A minha mãe e minha irmã que estão ao meu lado em qualquer desafio que eu venha me propor.

As minhas primas Van e Cris pelo enorme carinho com que me receberam em Salvador. Ao meu primo Allan pelos conselhos e ajudas que me concedeu sempre que precisei. Ao meu primo Dudu pelas parcerias que temos feitos nesses últimos tempos. As minhas tias Sônia e Liege: as melhores tias desse mundo. A minha madrinha Lúcia, uma das principais responsáveis pelas reuniões e alegrias da minha família. Ela nunca deixa o meu aniversário passar em branco. rs

Dentro da universidade também existiram pessoas que me ajudaram a concluir esta etapa. A começar pelo meu orientador Prof. Dr. Leobino Sampaio, primeiro professor da universidade que eu tive contato. Quero agradecê-lo pela confiança em mim depositada, pelo convívio amigável e a lealdade nas conversas.

Agradecer ao meu coorientador Prof. Dr. Gustavo Bittencourt pela seriedade, disponibilidade e dedicação a pesquisa científica. Agradecê-lo também pelas discussões realizadas durante toda essa caminhada, sem dúvida alguma foram de grande contribuição para a minha formação.

Agradecer ao colega Sérgio Gramacho, um cara que não deixa a pesquisa se distanciar da sua aplicabilidade. Isso é fundamental para qualquer pesquisador. Obrigado pelas conversas e companhia. Agradecer também ao meu colega e conterrâneo Cléber Araújo pelos bate-papos e incentivos.

Por fim, quero agradecer ao CNPq pelo financiamento desta pesquisa.

“O êxito na vida não se mede pelo que você conquistou,
mas sim pelas dificuldades que superou no caminho.”

—ABRAHAM LINCOLN

RESUMO

As redes de circuito dinâmico fazem parte de uma das tecnologias da Internet do Futuro e são caracterizadas por oferecer uma boa qualidade de serviço a um baixo custo. Através delas é possível aumentar a rapidez no estabelecimento dos circuitos, além de diminuir o custo de gerenciamento. A maioria dessas redes são voltadas para aplicações elásticas e tem os seus recursos alocados no modo de reserva antecipada. Entretanto, é muito provável que alocações de recurso no modo reserva imediata coexistam com as reservas antecipadas. Além disso, diferentes tipos de aplicações poderão ser suportadas, o que torna o problema de alocação de banda às requisições altamente desafiador e de fundamental importância para o seu bom desempenho.

Até onde se tem conhecimento, o único algoritmo de alocação de largura de banda da literatura possui dificuldades em permitir uma coexistência entre diferentes tipos de aplicações. Este trabalho concebe um novo algoritmo para alocação de banda em redes de circuito dinâmico denominado de Balanceamento de Banda Residual (BBR). O BBR explora características específicas de aplicações elásticas e procura balancear a banda residual dentro do intervalo de uma requisição. Adicionalmente se propõe uma estratégia de gerenciamento de recurso intitulada de Particionamento Dinâmico com Preempção com o objetivo de diminuir as diferenças de probabilidade de bloqueio entre as reservas antecipadas e as imediatas.

Experimentos através de simulação mostram que o BBR alcança uma menor probabilidade de bloqueio comparado à heurística comumente utilizada na literatura. Além disso, o Particionamento Dinâmico com Preempção diminui o impacto que as reservas antecipadas causam sobre as imediatas sem comprometer uma boa taxa de utilização da rede.

Palavras-chave: Redes de Circuito Dinâmico, Tráfego Elástico, Escalonamento de Largura de Banda, Reserva Antecipada, Reserva Imediata.

ABSTRACT

Dynamic circuit networks are part of one of the technologies of the Future Internet and are characterized by offering a good quality of service at a low cost. Through them it is possible to increase the speed in the establishment of the circuits, as well as lower the cost of management. Most of these networks are tuned to elastic applications and have their resources allocated in advance reservation. However, it is very likely that resource allocations in immediate reserve coexisted with advance reservation. In addition, different types of applications can be supported, which makes the problem of bandwidth allocation to highly challenging requests and of fundamental importance to its good performance.

To the best of our knowledge, the only algorithm of bandwidth allocation of literature have difficulties to allow a coexistence of different types of applications. This work conceives a new algorithm for bandwidth allocation in dynamic circuit networks called Band Residual balancing (BBR). The BBR explores specific features of elastic applications and seeks to balance the residual band within the range of a request. Additionally proposes a resource management strategy entitled Dynamic partitioning with Pre-emption in order to decrease the probability of blocking differences between the advance reservation and immediate.

Through simulation experiments show that BBR reaches a lower blocking probability compared to heuristic commonly used in the literature. In addition, the Dynamic Partitioning with Preemption decreases the impact that advance reservations about the immediate cause without compromise a good network utilization.

Keywords: Dynamic Circuit Network, Elastic Traffic, Scheduling Bandwidth, Advanced Reservation, Immediate Reservation.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Organização da Dissertação	3
Capítulo 2—Redes de Circuito Dinâmico	5
2.1 Reserva de Largura de Banda	9
2.2 Algoritmo Quick-Finish	12
2.2.1 Análise de Complexidade	14
2.3 Trabalhos Relacionados	16
Capítulo 3—O Algoritmo Balanceamento de Banda Residual	17
3.1 Análise de Complexidade	22
Capítulo 4—Experimentos Computacionais	25
4.1 Resultado para Diferentes Aplicações Inelásticas	26
4.1.1 Probabilidade de Bloqueio Total	26
4.1.2 Probabilidade de Bloqueio das Aplicações Inelásticas	29
4.1.3 Probabilidade de Bloqueio das Aplicações Elásticas	30
4.1.4 Taxa de Utilização da Rede	31
4.2 Resultados para Diferentes Flexibilidades	34
Capítulo 5—Gerenciamento Autônomo dos Recursos	39
5.1 Experimentos Computacionais	44
Capítulo 6—Considerações Finais	49
6.0.1 Trabalhos Futuros	50

LISTA DE FIGURAS

2.1	Modelo de Rede Híbrida	6
2.2	Topologia da rede ESnet, extraído de (VADREVU et al., 2010)	6
2.3	Interface Web do OSCARS para Realizar Login, extraída de (INTERNET2, 2013)	8
2.4	Interface web do OSCARS para submeter uma requisição, extraída de (INTERNET2, 2013)	9
2.5	Book-ahead time e o Horizon.	10
2.6	Problema Fragmentação da Largura de Banda.	10
2.7	Modelos de requisição, adaptada de (NETTO; BUBENDORFER; BUYYA, 2007).	11
2.8	Requisições já aceitas em cada caminho.	13
2.9	Requisição 7 alocada segundo o algoritmo QF em cada caminho.	15
3.1	Requisições já aceitas em cada caminho.	20
3.2	Requisição 7 alocada segundo o algoritmo BBR em cada caminho.	20
4.1	<i>Backbone</i> NSFNet	25
4.2	Probabilidade de Bloqueio Total em Função da Carga para os Modelos de Aplicação Inelásticas 1 à 4.	27
4.3	Tráfego de dados da rede de circuito dinâmico da ESnet, extraído de https://my.es.net/ , acessado em 21/02/2015.	28
4.4	Probabilidade de Bloqueio em Função da Carga para Diferentes Modelos de Reserva Imediata 5 à 8.	29
4.5	Probabilidade de Bloqueio das Aplicações Inelásticas em Função da Carga para os Modelos 1 à 4.	30
4.6	Probabilidade de Bloqueio das Reservas Imediatas em Função da Carga para os Modelos 5 à 8.	31
4.7	Probabilidade de Bloqueio das Reservas Antecipadas em Função da Carga para Diferentes Modelos de Reserva Imediata.	32
4.8	Probabilidade de Bloqueio em Função da Carga para Diferentes Modelos de Reserva Imediata.	33
4.9	Taxa de Utilização da Rede em Função da Carga para os Modelos de Aplicação Inelástica 1 à 4.	34
4.10	Probabilidade de Bloqueio em Função da Carga para os Modelos de Aplicação Inelástica 5 à 8.	35
4.11	Probabilidade de Bloqueio das Aplicações Elásticas em Função da Flexibilidade para Diferentes Cargas.	36

4.12	Probabilidade de Bloqueio das Aplicações Inelásticas em Função da Flexibilidade para Diferentes Cargas.	37
4.13	Taxa de Utilização da Rede em Função da Flexibilidade para Diferentes Cargas.	38
5.1	Diferentes Modos de Alocação de Recursos dentro de um Intervalo.	41
5.2	Distribuição dos Recursos entre Reserva Antecipada (RA) e Reserva Imediata (RI) em Função da Demanda.	41
5.3	Probabilidade de aceitação das reservas antecipadas para diferentes cenários utilizando diferentes estratégias de gerenciamento dos recursos.	46
5.4	Probabilidade de aceitação das reservas imediatas para diferentes cenários utilizando diferentes estratégias de gerenciamento dos recursos.	46
5.5	Taxa de utilização da rede para diferentes cenários utilizando diferentes estratégias de gerenciamento dos recursos.	48

LISTA DE TABELAS

2.1	Resumo sobre os diferentes tipos de aplicações.	11
2.2	Resumo sobre os diferentes tipos de reservas.	11
2.3	Resumo sobre os diferentes tipos de requisições.	12
4.1	Tabela com os parâmetros dos modelos 1 à 4 de RI	26
4.2	Tabela com os parâmetros dos modelos 5 à 8 de RI	28
5.1	Tabela com os diferentes cenários de tráfego.	44
5.2	Tabela com as cargas para diferentes demandas de RA e RI..	45

LISTA DE ACRÔNIMOS

ATM	Asynchronous Transfer Mode
AutoBAHN	Automated Bandwidth Allocation across Heterogeneous Networks
BBR	Balanceamento de Banda Residual
DCN	Dynamic Circuit Networks
ESnet	Energy Sciences Network
GMPLS	Generalized Multi-Protocol Label Switching
ISP	Internet Service Provider
JGN-X	Japan Gigabit Network Extreme
MPLS	Multi Protocol Label Switching
NSFNet	National Science Foundation Network
OSCARS	On-demand Secure Circuits and Advance Reservation System
QF	Quick-Finish
QoS	Quality of Service
REN	Research and Education Networks
RNP	Rede Nacional de Ensino e Pesquisa
RWA	Routing and Wavelength Assignment
SDN	Science Data Network
UCLP	User controlled LightPaths
WDM	Wavelength Ddivision Multiplexing

INTRODUÇÃO

Aplicações emergentes, como distribuição de conteúdo e *backup* com transferências massivas de dados, demandam grande largura de banda e possuem requisitos não compatíveis com o modelo de *melhor esforço* adotado na Internet. Para atender as exigências dessas aplicações foram criadas as DCN (Redes de Circuito Dinâmico, do Inglês *Dynamic Circuit Networks*), que oferecem um serviço confiável e eficiente, a partir de canais dedicados com grande largura de banda. A relevância desse serviço é percebida através da implantação dele em algumas das atuais REN (Redes de Ensino e Pesquisa, do Inglês *Research and Education Networks*), bem como em ISP (Provedor de Serviço de Internet, do Inglês *Internet Service Provider*) (CANARIE, 2013; INTERNET2, 2010; VEERARAGHA; JUKAN, 2010; JANET, 2013; SURFNET, 2013; EXTREME, 2013; RNP, 2013; ATT, 2013; VERIZON, 2013).

Por meio da DCN o usuário pode agendar sua requisição especificando a origem e o destino do circuito, a taxa de transmissão e o tempo de início e fim no uso dos recursos. Entretanto, esse tipo de requisição, denominado em (BURCHARD; HEISS; ROSE, 2003) de *requisição fixa*, favorece o problema de fragmentação da banda, que acontece quando o intervalo de tempo entre duas requisições é tão pequeno ao ponto de não ser possível alocar uma nova requisição, fazendo com que a largura de banda nesse intervalo seja desperdiçada. Para superá-lo, os autores propuseram o modelo de *requisição maleável*. Nesse modelo, ao invés dos usuários informarem exatamente o tempo de início e fim da transmissão, estes informam um intervalo dentro do qual seja aceitável realizar o escalonamento. Contudo, esse modelo de requisição não permite que a largura de banda seja variável, o que impede o aproveitamento de características específicas de algumas aplicações, dificultando um uso mais eficiente dos recursos. Foi sob esse fato que esta dissertação decidiu adotar o modelo de requisição chamado de *requisição elástica*, que é semelhante a *requisição maleável*, porém ainda permite variação na taxa de transmissão. No âmbito dos trabalhos que eliminam a necessidade da largura de banda ser fixa, a heurística mais comum utilizada pelos algoritmos de alocação de banda é o *Quick-Finish* (QF) (NAIKSATAM; FIGUEIRA, 2007; RAJAH; RANKA; XIA, 2009; PATEL et al.,

2009). Porém, essa heurística não é muito adequada para um ambiente em que coexistam aplicações com diferentes requisitos, por exemplo, sensível ao atraso ou sensível ao prazo de término, porque tende a criar instantes de tempo com completa ausência de largura de banda, acarretando o bloqueio de requisições que demandam uso imediato do recurso.

Diferentes tipos de aplicações poderão ser suportados pelas DCN. Por exemplo, a coexistência de aplicações elásticas com inelásticas. As aplicações elásticas não são sensíveis a fatores como atrasos, variação nos atrasos, flutuação na largura de banda e etc. Dentre as aplicações elásticas, as que mais se destacam são as aplicações “intensivas em dados” nas áreas de meteorologia, física de altas energias, transferência de arquivos em grades computacionais, etc. Por outro lado, as aplicações inelásticas são sensíveis a fatores como largura de banda, requerem uma taxa mínima de banda e certa latência. Esta heterogeneidade, gerada pela coexistência de diferentes tipos de aplicações, faz com que as soluções para os problemas de alocação de banda às requisições exerçam um papel fundamental para o bom desempenho da rede em acomodar novas aplicações. Por tais motivos, novas estratégias para o provisionamento de recursos em DCN são bastante oportunas, uma vez que aplicações elásticas possuem alta flexibilidade e, por isso, permitem muitas opções de escalonamento. O desafio está em definir qual dessas opções utilizar, de forma a permitir uma maior utilização dos recursos, minimizando, assim, as chances de bloqueio de requisições futuras.

Esta dissertação propõe um novo algoritmo denominado de Balanceamento de Banda Residual (BBR) para alocação de largura de banda em redes de circuito dinâmico. Esse algoritmo foi concebido para ser aplicado em um cenário onde coexistam aplicações elásticas com inelásticas. Adicionalmente se propõe uma estratégia de gerenciamento de recursos para ser implantada em uma rede de circuito dinâmico que permita ambos tipos de reserva: antecipada e imediata. Tal esquema tem como objetivo regular a distribuição de recursos entre esses tipos de reserva de maneira mais justa, sem deixar de ter uma boa utilização da rede.

Para avaliar o algoritmo foi construído um simulador de eventos discretos que simula a chegada das requisições, os diferentes requisitos de cada aplicação e os algoritmos de alocação de largura de banda. Experimentos comparando o algoritmo proposto com o algoritmo tradicionalmente utilizado na literatura mostraram que o BBR, devido a sua forma de alocação de banda, conseguiu aumentar de maneira significativa a aceitação das requisições com maiores restrições, o que ocasionou uma maior utilização da rede. Os resultados mostraram também que a estratégia de gerenciamento de recursos proposta conseguiu diminuir as diferenças entre os dois tipos de reservas utilizadas, tornando assim a distribuição dos recursos mais igualitária, ao mesmo tempo que fez uma utilização eficiente da rede.

Nesta dissertação serão apresentados os conceitos básicos sobre as redes de circuito dinâmico e os principais modelos de requisições e as suas características. Além disso, será discutido as duas principais formas de se realizar a reserva de recursos e os problemas que existem ao permitir a coexistência de ambos os modos.

1.1 ORGANIZAÇÃO DA DISSERTAÇÃO

A dissertação está organizada em seis Capítulos. No Capítulo 1, mostra-se o contexto da área pesquisada, o escopo e a relevância deste trabalho de pesquisa. No Capítulo 2, são apresentados os conceitos sobre redes de circuito dinâmico e os principais *frameworks* utilizados nessas redes. Na Seção 2.1, é explicado quais são os dois principais tipos de reserva de recurso, bem como os modelos de requisições que podem ser suportados pelas DCN. Na Seção 2.2, discute-se o funcionamento do algoritmo comumente utilizado para reservar recurso em redes de circuito dinâmico. Conclui-se o capítulo apresentando alguns trabalhos relacionados.

No Capítulo 3, a proposta desta dissertação é apresentada através da descrição do funcionamento do algoritmo “Balanceamento de Banda Residual”. No Capítulo 4, são expostos os resultados da simulação através das métricas de probabilidade de bloqueio e utilização da rede.

No Capítulo 5, discute-se o problema relacionado ao gerenciamento de recursos quando permitido a coexistência de reservas antecipadas com imediatas. Soluções amplamente conhecidas na literatura para tratar diferentes classes de requisições são analisadas. Uma adaptação dessas soluções foi implementado com intuito de obter melhores resultados. Por fim, no Capítulo 6 são feitas as considerações finais deste trabalho, contribuições realizadas e trabalhos futuros.

REDES DE CIRCUITO DINÂMICO

Uma rede de circuito dinâmico é aquela que oferece um *serviço* de transmissão e cria circuitos de curto prazo, com duração de minutos a dias, sob demanda, entre os usuários finais que necessitam de canais de largura de banda dedicada (INTERNET2, 2010). A maioria das redes que oferecem o serviço de circuito dinâmico, o fazem como complemento a sua rede IP tradicional, por exemplo, a Internet2 e a ESnet nos EUA oferecem esse tipo de serviço. Implantações similares também ocorrem na rede Européia GÉANT, e na asiática JGN-X (EXTREME, 2013) . Quando uma DCN coexiste com uma rede IP tradicional, tem-se a chamada *Rede Híbrida*.

Comumente, o serviço de DCN é oferecido em uma rede híbrida (INTERNET2, 2013; ESNET, 2013). Além disso, devido a complexidade operacional e as diferenças tecnológicas, os serviços IP e os serviços de DCN são provisionados através de duas redes logicamente e administrativamente separadas, como mostra a Figura 2.1. Esse é o caso, por exemplo, da ESnet (do Inglês *Energy Sciences Network*). A sua infraestrutura, mostrada na Figura 2.2, é composta de roteadores IP, responsáveis por oferecer o serviço tradicional IP, e de roteadores SDN (do Inglês *Science Data Network*) , responsável por oferecer o serviço de circuito dinâmico. Os equipamentos que constituem uma rede de circuito dinâmico vão desde switches WDM e ATM até switches de circuito virtual como o switch Ethernet e o MPLS .

O serviço de rede de circuito dinâmico introduzido por REN's e ISP's veio para preencher o espaço existente entre os serviços de comunicação amplamente usados atualmente como o serviço IP e o serviço de linha alugada. O serviço de linha alugada se caracteriza por oferecer canais com grande largura de banda, sendo os nós origem e destino, a quantidade de banda e a duração, todos, definidos de uma só vez no ato do contrato com o prestador de serviço. Além disso, normalmente, os nós de origem e destino não podem ser alterados, a duração mínima do circuito costuma ser de um ano e possui um alto custo para aquisição. No outro extremo, tem-se os serviços IP, caracterizado por ser um serviço sem conexão, o qual implica que recursos não são alocados previamente, e possuindo um alto nível de compartilhamento dos recursos. Já os serviços de circuito dinâmico não

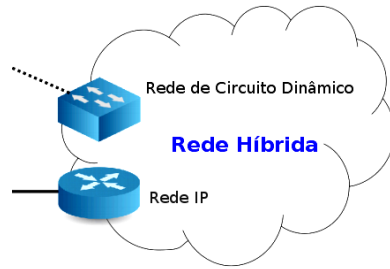


Figura 2.1: Modelo de Rede Híbrida

possuem o nível de compartilhamento do serviço IP, mas também não possui a rigidez do serviço de linha alugada. A DCN se caracteriza por ser bastante flexível, dando ao usuário a opção de escolher quaisquer nós que façam parte da DCN para ser o nó origem e destino para uma dado tráfego, bastando apenas que o usuário esteja conectado a um dos switches da DCN. A duração pode ser de minutos a dias, dependendo da DCN, e os custos são inferiores aos serviços de linha alugada. Por isso, as DCN se caracterizam por oferecer uma boa qualidade de serviço a um baixo custo.

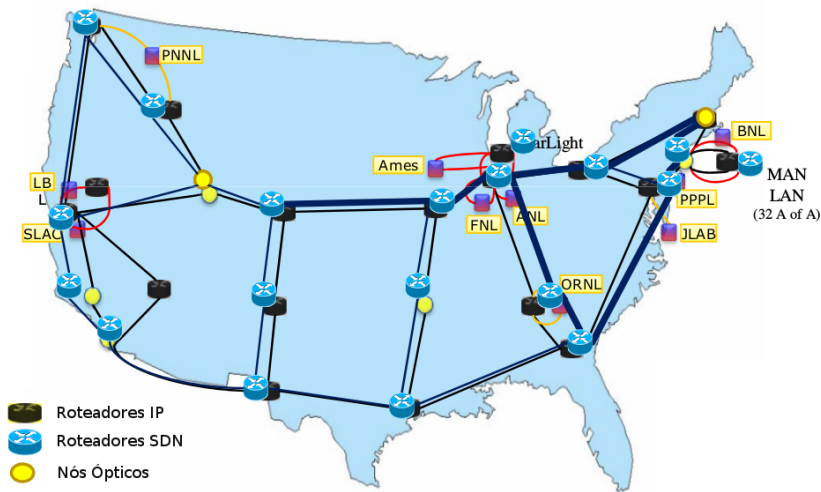


Figura 2.2: Topologia da rede ESnet, extraído de (VADREVU et al., 2010)

Para dar suporte ao estabelecimento de canais dedicados nessas redes que oferecem circuitos dinâmicos, *frameworks* são desenvolvidos para implementarem o plano de controle, como é o caso do OSCARS (OSCARS, Janeiro, 2014), amplamente difundido e implantado em redes que oferecem esse tipo de serviço. Outros exemplos são o AUTOBAHN (AUTOBAHN, 2013), o UCLP (UCLP, 2013) e outros.

Para entender os tópicos tratados por esses frameworks, será descrito de maneira sucinta os componentes que fazem parte do OSCARS (do Inglês On-demand Secure Circuits and Advance Reservation System), que é o software utilizado, por exemplo, na

Rede Nacional de Ensino e Pesquisa (RNP) .

O OSCARS é o responsável por criar circuitos com grande capacidade de largura de banda, dinamicamente, de forma segura, atendendo as restrições dos usuários. As várias funcionalidades do OSCARS são fornecidas pelos oito componentes (ESNET, 2015) aqui descrito resumidamente:

- ***Web-browser interface/programmatic API***: Esse módulo interage diretamente com a aplicação do usuário para obter informações sobre os requisitos de conectividade tais como a quantidade de largura de banda, a duração da conexão e os pontos de origem e destino.
- ***Authentication/Authorization/Accounting module***: Autentica o usuário e autoriza a solicitação do mesmo. Também fornece informações para auditoria.
- ***Topology manager***: Mantém e atualiza as informações do *link* da topologia de rede para que o módulo *Path-computation engine* possa calcular com precisão os caminhos da solução.
- ***Resource manager***: Mantém informações para acompanhar os vários estados(reservado, ativo, cancelado ou concluído) de cada reserva. Em essência, o gerente de recursos é um banco de dados que armazena todas as informações sobre cada reserva.
- ***Notification broker***: Notifica o usuário ou qualquer outra entidade se o estado das reserva foi alterado. A notificação pode tomar a forma de mensagens de serviço da Web, e-mails, e mensagens de texto.
- ***Path-computation engine***: Determina se um caminho de solução que atenda a todas as restrições especificadas pelo usuário existe dentro do gráfico ligação topologia da rede.
- ***Device drivers***: Comunica-se diretamente com os elementos da rede (roteadores, switches, etc.) para configurar e remover os circuitos conforme especificado pelo módulo *Path-computation engine*, segundo a duração e largura de banda solicitada pelo usuário. Este componente compreende tecnologias específicas da plataforma de fornecedores (GMPLS , MPLS, Ponte Ethernet, Qualidade de Serviço). Essa capacidade de abstrair o hardware da rede subjacente é um aspecto fundamental para tornar o OSCARS independente de tecnologia.
- ***Workflow coordinator***: Coordena o fluxo de trabalho dos outros componentes e solicitações de usuários em vários domínios de rede para facilitar o processo de reserva de rede.

Um usuário para tentar realizar uma reserva, primeiramente deverá acessar a interface Web e passar por uma etapa de autenticação, onde é verificado se o usuário tem permissão para realizar tal reserva, como mostra a Figura 2.3.

Nesse formulário, dois campos devem ser preenchidos:

December 19, 2011 12:58 [Log in to OSCARS.](#)

Login/Logout

User Name:

Password:

Sign in via your OSCARS login and password to access this system. To find out about OSCARS and how it works, go to the [documentation](#). To obtain an account or request more information, email one of the contacts below.

More documentation can be found by clicking on the link at the bottom of this page.

Figura 2.3: Interface Web do OSCARS para Realizar Login, extraída de (INTERNET2, 2013)

- *User Name*: Campo em que deve ser informado o nome do usuário que deseja acessar o sistema.
- *Password*: Campo que deve receber a senha do usuário.

Caso o usuário possua as credenciais necessárias, este poderá realizar um agendamento de um circuito de acordo com a sua exigência, como mostra a Figura 2.4. Na etapa de solicitação do circuito existe uma série de campos a serem preenchidos, dentre eles os principais são:

- *Source*: Campo que deve ser preenchido com o nó de origem do circuito.
- *Destination*: Campo que deve ser preenchido com o nó de destino do circuito.
- *Bandwidth*: Largura de banda requerida para o circuito, especificado em Mpbs.
- *Start Date*: Data para o início do estabelecimento do circuito.
- *Start Time*: Hora para o início do estabelecimento do circuito.
- *End Date*: Data para o encerramento do circuito provisionado.
- *End Time*: Hora para o encerramento do circuito provisionado.

O gerenciador de recurso, com base nos recursos disponíveis e no circuitos já agendados, irá definir se é possível ou não atender a requisição. Após essa decisão, o usuário será notificado sobre o resultado do circuito demandado. Caso a requisição tenha sido aceita, um módulo específico ficará responsável por configurar os dispositivos da rede para atender a respectiva requisição.

Esta dissertação defende o modelo de *requisição elástica*, no qual o usuário não informa exatamente o momento em que o circuito será provisionado e removido, mas sim um intervalo de tempo aceitável para que a aplicação conclua a sua transmissão. A tarefa de definir o momento exato em que o circuito será estabelecido passa a ser do gerenciador do recurso, pois ele tem o conhecimento sobre a disponibilidade dos recursos.

INTERNET2 ESnet

On-demand Secure Circuits and Advance Reservation System

A collaboration between ESnet, Internet2, DANTE, and ISI East

June 4, 2008 21:22

Reservation creation form

Reservations | Reservation Details | Create Reservation | User List | Add User | User Profile | Login/Logout

Required inputs are bordered in green. The source and destination can be topology identifiers, host names, or IP addresses, depending on the layer used. Click on the boxes associated with the start and end dates to bring up a calendar widget. The reservation time slot defaults to now, and now + 4 minutes, respectively, if you leave the dates and times empty.

WARNING: Entering a series of hops in the Path field may alter routing behavior for the selected flow. Hops can be topology identifiers, host names, or IP addresses, depending on the layer used. Note that the path field will expand to the number of lines occurring in the hops list.

Create Reservation Production circuit

Source: phoebus.losa.dcn.internet2.edu

Destination: phoebus.chic.dcn.internet2.edu

Path (series of hops):

Bandwidth (Mbps): 1,000 (10-10000)

Description: test (For our records)

Start date: 6/4/2008

Start time: 21:22

End date: 6/4/2008

End time: 21:26

Use layer 2 parameters Use layer 3 parameters

VLAN: tag, or range, e.g. 3000-3100

Figura 2.4: Interface web do OSCARS para submeter uma requisição, extraída de (INTERNET2, 2013)

2.1 RESERVA DE LARGURA DE BANDA

A reserva de recursos é uma maneira comum de prover QoS (do Inglês *Quality of Service*) para aplicações. Existem dois tipos de reserva: reservas imediatas (RI) no qual os recursos são reservados imediatamente após a chegada da requisição, e as reservas antecipadas (RA) em que os recursos são reservados após um dado intervalo desde a chegada da requisição. A RI é a forma tradicional de reserva em redes de circuito como é o caso da rede de telefonia. Nesse tipo de reserva, a alocação do recurso coincide com a chegada da requisição. Já na RA, o recurso é alocado não no momento em que a requisição chega, mas em um tempo futuro.

As reservas antecipadas têm sido amplamente implementadas nas DCN (CHARBONNEAU et al., 2011). Um dos motivos para a maioria das DCN adotarem a RA é devido ao fato desse modo de alocação diminuir a probabilidade da requisição ser bloqueada. Isso acontece, pois normalmente as aplicações destinadas ao modo RA demandam muito recurso, e quanto menor for a antecedência da reserva, maiores são as chances destes estarem sendo utilizados por outras requisições.

Para implementar o sistema de reserva antecipada, é necessário definir o intervalo de tempo entre a chegada da requisição e o início do uso dos recursos. Esse intervalo é conhecido como *book-ahead time* (WOLF; STEINMETZ, 1997). Os sistemas que utilizam a reserva antecipada normalmente definem um valor mínimo e máximo para o *book-ahead time*. O valor mínimo é um modo de evitar que o intervalo entre a chegada da requisição e o uso dos recursos seja tão curto a ponto de não ter diferença para uma RI. O valor máximo é para prevenir um armazenamento muito grande de estados das requisições no sistema.

Esse valor máximo normalmente é referido como *horizon* (ZHU; VEERARAGHAVAN, 2008). Esses elementos podem ser vistos na Figura 2.5.

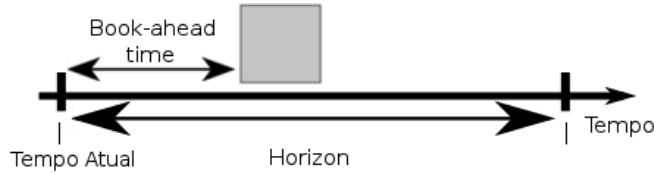


Figura 2.5: Book-ahead time e o Horizon.

A forma tradicional de realizar uma reserva antecipada é através da *requisição fixa*, que conforme mencionado anteriormente contribui para o problema de fragmentação da largura de banda, como mostra a Figura 2.6.

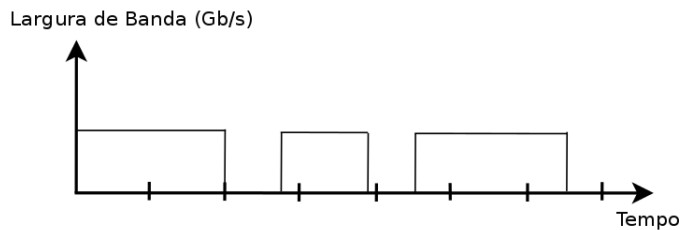


Figura 2.6: Problema Fragmentação da Largura de Banda.

Na *requisição fixa*, o usuário define todos os parâmetros necessários para a aplicação, como por exemplo, o tempo de início e fim da transmissão e uma taxa de transmissão constante, como mostra a Figura 2.7a. O conceito de *requisição maleável* proposto por Burchard et al. em (BURCHARD; HEISS; ROSE, 2003) trata o problema da fragmentação da banda. Nesse novo modelo de requisição, ao invés de especificar exatamente o tempo de início e fim no uso dos recursos, especifica-se um intervalo dentro do qual a requisição precisa ser satisfeita, como mostra a Figura 2.7b. Entretanto, requisições maleáveis não permitem o uso de largura de banda variável.

As *requisições elásticas*, por sua vez, são aquelas que possuem as características das requisições maleáveis adicionando a possibilidade de a largura de banda ser variável (NAIK-SATAM; FIGUEIRA, 2007). Embora o modelo de *requisição elástica* não possua restrições fixas de tempo de início, nem uma taxa de transmissão constante, esse modelo tem restrição com relação ao prazo de término da transmissão que precisa ser atendido. Esse tipo de requisição permite ao algoritmo de alocação de banda a possibilidade de preencher instantes do tempo em que não há qualquer alocação de recursos, minimizando assim a fragmentação da largura de banda.

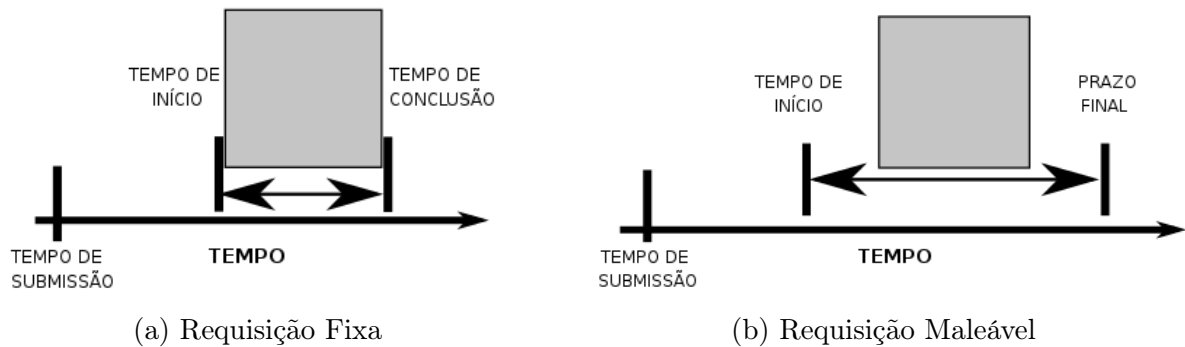


Figura 2.7: Modelos de requisição, adaptada de (NETTO; BUBENDORFER; BUYYA, 2007).

Como é possível observar na Figura 2.7b, a requisição possui um tempo de início e fim no qual a transmissão dos dados precisa ser concluída. Dentro desse intervalo, o algoritmo de alocação de banda pode aproveitar as características específicas, por exemplo, de aplicações elásticas, como é o caso da taxa de transmissão poder ser variável. As Tabelas 2.1, 2.2 e 2.3 mostram de forma resumida os principais conceitos desta dissertação discutidos até o momento.

Tabela 2.1: Resumo sobre os diferentes tipos de aplicações.

Tipo de Reserva	Resumo
Antecipada	Existe uma desacoplamento entre a chegada da requisição e o momento em que o recurso é alocado. Normalmente esse tipo de reserva é planejado com antecedência. É a forma de reserva utilizada pela maioria das redes de circuito dinâmico.
Imediata	A alocação do recurso coincide com a chegada da requisição. É o tipo de reserva que não foi planejado antecipadamente. Ainda pouco utilizado nas redes de circuito dinâmico.

Tabela 2.2: Resumo sobre os diferentes tipos de reservas.

Tipo de Aplicação	Resumo
Elástica	São aplicações não sensíveis, por exemplo, a variações na largura de banda.
Inelástica	São aplicações que requerem um quantidade mínima de largura de banda para funcionar.

Para qualquer transferência ser realizada existe um tempo mínimo para a sua conclusão, que é baseado na máxima capacidade de rede. Por exemplo, para transferir um arquivo de 2.5 GB em uma rede que suporta no máximo 10 Gbps, são necessários no mínimo 2 segundos. Quanto maior for o intervalo de uma *requisição elástica* comparado

Tabela 2.3: Resumo sobre os diferentes tipos de requisições.

Tipo de Requisição	Resumo
Fixa	O dono da aplicação define o tempo de início e fim da transmissão e a taxa da largura de banda.
Flexível	O dono da aplicação define um intervalo dentro do qual seja aceitação concluir a transmissão. A taxa de transmissão é decidida pelo gerenciador do recurso e ela permanece constante durante toda a transmissão.
Maleável	Semelhante a requisição flexível, porém permitindo que a largura de banda seja variável.

a esse tempo mínimo, maior é a sua *flexibilidade*. Para melhor exemplificar o conceito de flexibilidade, suponha que uma dada aplicação de transferência massiva de dados necessite transmitir 1 TB de dados em uma rede que suporta 100 Gbps como é o caso da ESnet. Essa rede precisa de no mínimo 80 segundos para concluir a transmissão. Isso na melhor das hipóteses, consumindo toda a banda disponível no caminho. Quando se diz que a flexibilidade de uma requisição é 1, significa que o tempo para concluir a transmissão é igual ao menor tempo possível. Por outro lado, quando se diz que a requisição possui flexibilidade 2, significa que a requisição permite que a mesma seja concluída em até duas vezes a duração mínima. Quanto maior for a flexibilidade, em teoria, maiores são as chances de a requisição ser aceita no sistema.

2.2 ALGORITMO QUICK-FINISH

Uma heurística bastante utilizada pelos algoritmos de alocação de largura de banda que permitem a largura de banda ser variável é o *Quick-Finish* (QF) (NAIKSATAM; FIGUEIRA, 2007; RAJAH; RANKA; XIA, 2009; PATEL et al., 2009). Essa heurística tem como objetivo concluir o mais cedo possível a transmissão de uma dada requisição.

Uma *requisição elástica* é representada por $R = (n^s, n^d, t^I, t^F, D)$, em que n^s e n^d referem-se aos nós de origem e de destino do circuito, respectivamente; t^I representa o instante permitido para iniciar a transmissão, t^F é o tempo máximo para finalizar a transmissão e, por fim, D representando a demanda da requisição representada através de *slots de largura de banda*. Vale destacar que a flexibilidade está embutida no intervalo $[t^I, t^F]$. Assim, a flexibilidade não é um parâmetro visível nos argumentos dos algoritmos.

A heurística *Quick-Finish* está representada no Algoritmo 1. A seguir, são apresentadas as variáveis utilizadas pelo algoritmo e suas respectivas descrições:

- **k** - Número de menores caminhos utilizados pelo algoritmo.
- **path** - Representa um dos k menores caminhos.
- **timeSlotIndex** - Representa o slot corrente durante uma busca dentro de um intervalo.
- **maxBandwidth** - Largura de banda máxima obtida em um dado caminho.

- **L** - Armazena caminhos com os valores de banda alocados em cada instante e o instante final em que a transmissão se concluirá em cada caminho.

Esse algoritmo inicia calculando os k -menores caminhos entre os nós origem e destino (linha 1). Para cada caminho, inicia-se a busca por recursos a partir do instante t^I até o t^F (laço da linha 2). Em cada instante, o algoritmo busca a máxima quantidade de banda disponível (linha 5). A máxima banda aqui conseguida é limitada pelo *link* com a menor capacidade no caminho.

Essa busca continua nos instantes seguintes (linhas 4-14), até a quantidade de banda ser suficiente para atender a demanda da requisição ou até alcançar o prazo máximo de término da mesma. Caso a demanda seja atendida, o algoritmo executa as instruções contidas nas linhas 9-11.

Ao final do laço que percorre o intervalo da requisição (linhas 4-14) é necessário verificar por qual motivo o laço foi encerrado. Se o laço foi encerrado porque o caminho que estava sendo analisado satisfaz a requisição (linha 15), o algoritmo executará as instruções contida nas linhas 16-17. Caso o laço tenha sido encerrado porque o tempo máximo para concluir a transmissão foi alcançado, o algoritmo executa a instrução contida na linha 19. Esse processo é repetido até que todos o k menores caminhos sejam analisados.

Por fim, ao sair do laço mais externo (linhas 2-21) é verificado se algum dos k menores caminhos satisfaz a requisição (linha 22). Caso mais de um caminho possa atender a requisição, escolhe-se o que conclui a transmissão no tempo mais cedo (linha 23).

Exemplificando o funcionamento do algoritmo, imagine que em um sistema, cujo domínio do tempo foi dividido em *slots*, e em cada *slot* seja possível alocar até 5 Gbps de largura de banda, abstraídos em *slots de largura de banda* de 1 Gbps cada. Além disso, existem apenas 2 caminhos entre um específico nó de origem e destino, sendo cada um deles com três requisições já aceitas. Essas requisições estão enumeradas de 1 à 6 como mostra a Figura 2.8a e 2.8b. O eixo x representa a linha do tempo e o eixo y os slots de largura de banda. A requisição 1, por exemplo, consome 2 slots de largura de banda, enquanto que as requisições 2 e 3 consomem 3 slots de largura de banda.

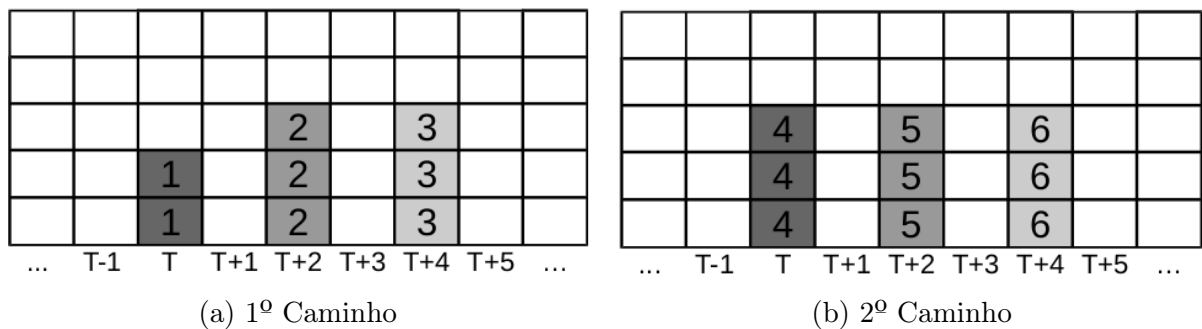


Figura 2.8: Requisições já aceitas em cada caminho.

Suponha que uma nova requisição enumerada de 7 chegue a esse sistema com os intervalos para início e término sendo iguais à T e $T+5$ e que demande 10 Gbps.

Algoritmo 1: *Quick-Finish* (QF)**Entrada:** $R = (n^s, n^d, t^I, t^F, D)$ **Saída:** aceita ou rejeita a requisição

```

1 for path = 0 to path < K do
2   timeSlotIndex = tI;
3   while timeSlotIndex <= tF do
4     maxBandwidth = MaxBandwidth(path, timeSlotIndex);
5     if D > maxBandwidth then
6       D = D - maxBandwidth;
7     else
8       maxBandwidth = maxBandwidth - D;
9       D = 0;
10      break;
11     end
12     timeSlotIndex = timeSlotIndex + 1;
13   end
14   if D == 0 then
15     A requisição nesse path conclui no slot representado em timeSlotIndex;
16     Insere o path em uma lista de soluções L;
17   else
18     Esse path não possui recursos;
19   end
20 end
21 if Houver caminho em L then
22   Escolhe aquele que conclui o mais cedo a transmissão;
23 else
24   rejeita a requisição;
25 end

```

Conforme mostra a Figura 2.8a, apenas no instante T , que é o primeiro instante do intervalo da requisição, o caminho 1 possui mais recursos que o segundo caminho, Figura 2.8b. Porém, esse primeiro instante é suficiente para fazer com que a requisição no caminho 1 termine no instante $T+2$, como mostra a Figura 2.9a e, por isso, seja o caminho escolhido. Uma vez que no segundo caminho a requisição só concluiria no instante $T+3$, como pode ser visto na Figura 2.9b.

2.2.1 Análise de Complexidade

Para o cálculo dos k menores caminhos, utilizou-se o algoritmo proposto por (YEN, 1971). Esse algoritmo utiliza qualquer algoritmo de caminho mínimo para encontrar o menor caminho e, então, a partir dele, obtém os outros k menores caminhos. Esse algoritmo possui complexidade cúbica e, por ser utilizado no modo *offline*, além de compartilhado

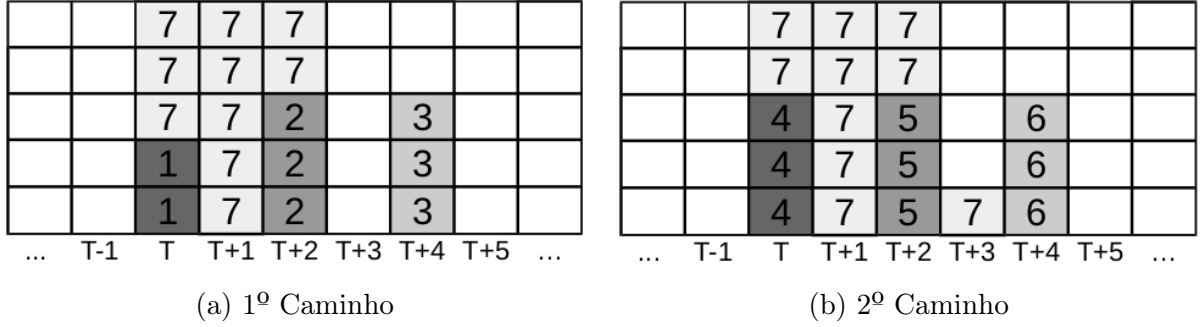


Figura 2.9: Requisição 7 alocada segundo o algoritmo QF em cada caminho.

pelos algoritmos *Quick-Finish* e Balanceamento de Banda Residual, não será levado em consideração na complexidade final do algoritmo.

A partir da linha 2 do Algoritmo 1 é que de fato se inicia as características específicas do mesmo, as quais são responsáveis pelo controle de admissão da requisição no modo *online*. O laço mais externo que controla a verificação dos caminhos (linhas 2-21) é executado k vezes. Dentro desse laço, existe outro laço (linhas 4-14) responsável por percorrer o intervalo de uma dada requisição. A duração desse intervalo é representada por $d_{interv} = t^F - t^I$. Nem sempre é necessário percorrer todo o intervalo. É possível que a requisição seja satisfeita sem que o intervalo inteiro seja analisado. No pior caso todo o intervalo é verificado. Até aqui, tem-se que a complexidade do algoritmo é:

- Pior Caso: $O(kd_{interv})$

Em cada iteração controlada pelo laço (4-14) é executado a função *MaxBandwidth*, que retorna a máxima largura de banda em um caminho específico. Como o algoritmo proposto em (YEN, 1971) é sem “loops”, sabe-se que qualquer que seja um dos k menores caminhos utilizados pela função *MaxBandwidth*, ele não será maior que o número de vértices. Sendo $|V|$ o número total de vértices em uma topologia de rede, a complexidade desta função será $O(|V|)$. Assim, o Algoritmo 1 passa a ter complexidade:

- Pior Caso: $O(kd_{interv}|V|)$

Ao final, as instruções contidas entre as linhas (22-24), que não é uma operação dominante do algoritmo, será executada apenas caso haja algum caminho com recurso. A complexidade dessa operação é $O(k)$, no pior caso, situação em que todos os k menores caminhos possam satisfazer a requisição. No caso médio apenas $k/2$ caminhos serão analisados. Portanto, a complexidade final do Algoritmo 1 é:

- Pior Caso: $O(kd_{interv}|V| + k)$

Quanto maior for o intervalo da requisição a ser percorrido, maior é o esforço computacional desse algoritmo. Portanto, a medida que a rede passa a ter mais reservas aceitas, o algoritmo tende a percorrer uma maior parte deste intervalo, uma vez que os recursos se tornam mais escassos.

2.3 TRABALHOS RELACIONADOS

Em (BALMAN et al., 2010), os autores assumem utilizar requisições maleáveis para agendar larguras de banda sem que seja permitido que a mesma possa variar. Consequentemente, isto impede o aproveitamento das características específicas de algumas aplicações, de modo a melhorar a eficiência na utilização dos recursos.

Em uma evolução do trabalho desenvolvido, em (BALMAN, 2013) permite-se que requisições já aceitas possam ser realocadas para admitir novas requisições. A realocação consiste em analisar cada requisição já alocada dentro do intervalo de tempo da nova requisição, e verificar se é possível rearranjá-la de tal forma que se garanta que as requisições já aceitas somadas a requisição mais recente possam ser admitidas no sistema, sem que nenhuma restrição das mesmas sejam violadas. O problema com essa abordagem é a questão de escalabilidade, uma vez que o reescalonamento é feito no modo *online*.

Em (RAJAH; RANKA; XIA, 2009), os autores utilizam a heurística QF com e sem reescalonamento online. Em (NAIKSATAM; FIGUEIRA, 2007), os autores propuseram um algoritmo que inicialmente faz a alocação da banda utilizando a heurística QF. Quando esse modo de alocação falha, permite-se fazer um reescalonamento online nas requisições já admitidas com o intuito de aceitar a nova requisição. O problema desse algoritmo é a escalabilidade, além de ter sido apresentado em uma rede com um único *link*. Nota-se em (PATEL et al., 2009) que os autores também utilizam a heurística QF e não permitem o reescalonamento. Todavia, a requisição não tem um prazo para concluir a transmissão.

Alguns trabalhos também permitem fazer realocações de requisições no modo *offline* (SCHMIDT; KUNEGIS, 2007; SHEN et al., 2007). Isso significa que uma requisição já agendada possa passar por um novo reescalonamento. Em (SCHMIDT; KUNEGIS, 2007), os autores permitem fazer mudanças apenas no caminho da requisição, sendo transparente para o usuário. Contudo, esse trabalho não aborda o uso de largura de banda variável. Em (SCHMIDT; KUNEGIS, 2007; SHEN et al., 2007) o foco é voltado para o problema de roteamento e atribuição de comprimento de onda.

Em (SHARMA; KATRAMATOS; YU, 2011), as requisições também são maleáveis, porém as mesmas não são acomodadas uma a uma de acordo com a ordem de chegada, mas todas de uma única vez. Diferentemente dessa abordagem, este trabalho utiliza um modelo mais realista no qual não se tem conhecimento das requisições futuras. Em (PATEL; JUE, 2011) já é permitido o uso da largura de banda variável, porém as requisições não possuem prazo para concluir a transmissão.

Dentre todos os trabalhos que se tem conhecimento da literatura que permite que a banda seja variável, a heurística geralmente utilizada é geralmente: concluir a transmissão o mais cedo possível. Essa heurística naturalmente é aplicada para tráfegos elásticos. Para satisfazer esse princípio, alocações máximas de banda são realizadas. Esta dissertação defende que essa abordagem não é muito adequada quando coexistem aplicações elásticas e inelásticas, pois devido as exigências das aplicações inelásticas serem mais rígidas, estas, por sua vez, não se adaptam facilmente a condição de rede deixada pela forma como atualmente são alocadas as aplicações elásticas na redes de circuito dinâmico.

O ALGORITMO BALANCEAMENTO DE BANDA RESIDUAL

Aplicações inelásticas se caracterizam por possuírem uma série de restrições para o seu bom funcionamento, como, por exemplo, uma largura de banda mínima. Um bom exemplo de aplicação inelástica são as aplicações do tipo *streaming*. Esse tipo de aplicação tem dificuldades para ser aceita em uma rede onde a alocação de recursos é feita segundo a heurística *Quick-Finish*. Isto acontece porque esse modo de escalonamento tende a criar muitas variações de banda residual no tempo. Isso implica em encontrar instante sem nenhum recurso até instantes com completa oferta de recurso. Quanto maior for essa variabilidade, tem-se como consequência, a dificuldade no atendimento de requisições com restrição de banda mínima, como é o caso das aplicações de streaming. O algoritmo de Balanceamento de Banda Residual (BBR), proposta para aplicações elásticas, por sua vez, consiste em buscar balancear a banda residual no intervalo de uma dada requisição. Entre seus objetivos, está o aumento na probabilidade de aceitação de aplicações inelásticas.

Por balanceamento, entenda-se deixar os instantes de tempo com a mesma banda residual. Além disso, uma variabilidade muito grande nos recursos ao longo do tempo dificulta a atribuição de banda em períodos contínuos. Por esse motivo, o algoritmo BBR tem o intuito de fazer um esquema de balanceamento que procura alocar mais recursos no instantes com maior disponibilidade, e menos, no que tiver menor disponibilidade. Através dessa estratégia, o BBR evita que em determinados períodos a banda seja exaurida e, conseqüentemente, bloqueie as aplicações inelásticas. O funcionamento do BBR é descrito formalmente através dos Algoritmos 2 e 3. A seguir, são apresentadas as variáveis utilizadas pelos algoritmos:

- **k** - Número de menores caminhos utilizados pelo algoritmo.
- **path** - Representa um dos k menores caminhos.
- **timeSlotIndex** - Representa o slot corrente durante uma busca dentro de um intervalo.

- **sum** - Somatório de larguras de banda.
- **L** - Armazena caminhos com os valores de banda alocados em cada instante e a diferença entre a máxima e mínima banda residual para cada caminho.

Inicialmente, o algoritmo computa os k menores caminhos entre os nós origem e destino (Linha 1 do Algoritmo 2). Em seguida, é calculada a máxima capacidade de banda para cada instante dentro do intervalo $[t^I, t^F]$ utilizando a função *MaxBandwidth* (Linhas 3 à 5 do Algoritmo 2). A partir daí, tem-se o somatório com todas as larguras de banda disponíveis para o intervalo da requisição corrente. Com esses valores é possível decidir se a rede pode ou não conseguir comportar tal requisição. Caso sim, o algoritmo irá executar as instruções contidas entre as linhas 8 e 11, responsável por atribuir, de forma proporcional, a banda nesses instantes. O objetivo com essa alocação é uniformizar a banda residual no intervalo $[t^I, t^F]$, que se refere ao tempo mais cedo para iniciar a transmissão de uma dada requisição e o tempo máximo para finalizá-la. Ao final, nem sempre será possível atribuir a demanda da requisição de forma que todos os instantes, dentro do intervalo da requisição, possuam a mesma banda residual. Isso irá depender de como se encontra a banda residual no intervalo de tempo da requisição corrente e dos requerimentos da própria requisição.

Ao final do laço referente a análise de cada caminho (linhas 2-15), o algoritmo verifica se algum caminho conseguiu satisfazer a requisição (linha 16). Caso sim, a função *Seleção de Caminho* é executada. Nesta função, retorna-se o caminho em que, uma vez alocada a requisição, o algoritmo o fez de modo a deixar a menor variação da banda residual no intervalo da requisição.

Assim, uma vez escalonada a requisição em diferentes caminhos, é provável que cada caminho, dentro do intervalo da requisição, possua banda residual distintas entre eles. Consequentemente, ao final de um escalonamento, teremos uma balanceamento da banda residual diferente para cada caminho. A política de seleção de caminho procura selecionar o caminho com o melhor balanceamento nesse intervalo, isto é, o que possuir menor variação de disponibilidade de recurso após a alocação da requisição. Quando existe uma mesma variação em mais de um caminho, escolhe-se o que tiver a maior média de banda residual, pois é o caminho que possui mais recursos dentro do intervalo da requisição.

Assumindo o mesmo exemplo de estado da rede e requisições dado no Algoritmo 1, tem-se o caminho 1 com o estado dos recursos representado pela Figura 3.1a e o caminho 2 representado pela Figura 3.1b.

As Figuras 3.2a e 3.2b já mostram o resultado do escalonamento usando o BBR para o primeiro caminho e o segundo caminho, respectivamente.

Como pode ser visto, o BBR procura dentro do intervalo de tempo da requisição, nesse caso específico $[T, T+5]$, fazer um balanceamento da banda residual. Com isso, os instantes $T+2$ e $T+3$, que possuem a menor banda residual, foram os que menos alocações tiveram, ao passo que os instantes $T+1$, $T+3$ e $T+5$ foram os que sofreram mais alocações, por não terem nenhum recurso alocado neles no momento da chegada da requisição.

Essa abordagem, embora atrase o fim da transmissão, tende a adiar o exaurimento da largura de banda nos enlaces em cada instante do tempo, além de favorecer uma maior

Algoritmo 2: BBR

Entrada: $R = (n^s, n^d, t^I, t^F, D)$ **Saída:** aceita ou rejeita a requisição

```

1 for  $path = 0$  to  $path < K$  do
2    $sum = 0$ ;
3   for  $timeSlotIndex = t^I$  to  $timeSlotIndex \leq t^F$  do
4      $sum += \text{MaxBandwidth}(path, timeSlotIndex)$ ;
5   end
6   if ( $sum \geq D$ ) then
7     for  $timeSlotIndex = t^I$  to  $timeSlotIndex \leq t^F$  do
8       Aloca em  $timeSlotIndex$  de forma proporcional a sua disponibilidade
       de recursos;
9     end
10    Insere o  $path$  em uma lista de soluções  $L$ ;
11  else
12    Esse  $path$  não possui recursos;
13  end
14 end
15 if Houver caminho em  $L$  then
16   Seleção de Caminho (Algoritmo 3);
17 else
18   rejeita a requisição;
19 end

```

Algoritmo 3: Seleção de Caminho

Entrada: Um ou mais caminhos entre n^s e n^d **Saída:** Um único caminho para o destino

```

1 if Houver mais de um caminho para alcançar o destino then
2   Escolha o caminho que tiver a menor diferença entre a máxima e a mínima
   banda disponível no intervalo da requisição;
3   if Se houver mais de uma opção then
4     Escolha o caminho que tiver a maior média baseada na banda disponível
     em cada  $slot$  dentro do intervalo da requisição;
5   else
6     Escolhe o primeiro da lista
7   end
8 else
9   Retorna o único caminho obtido;
10 end

```

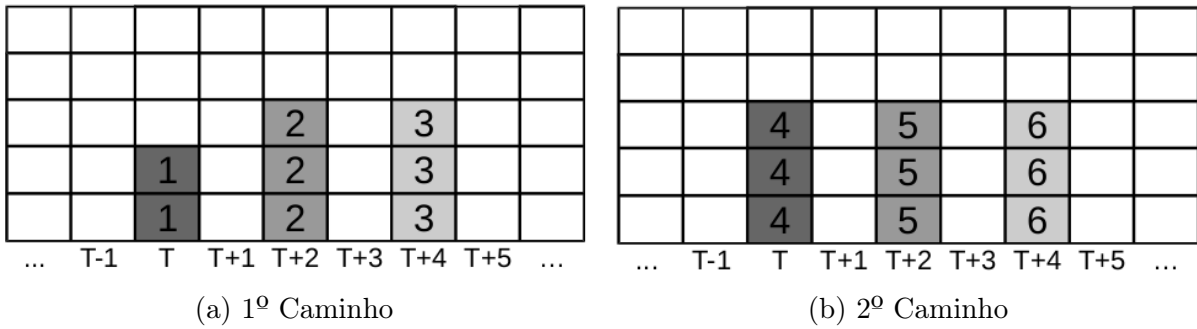


Figura 3.1: Requisições já aceitas em cada caminho.

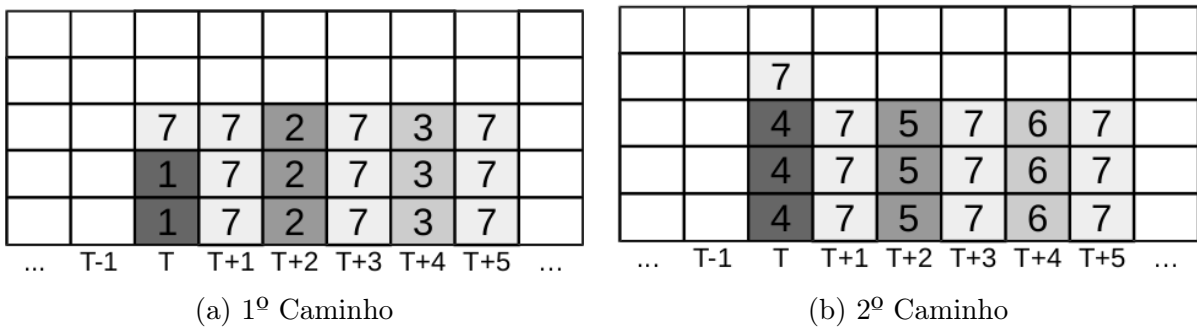


Figura 3.2: Requisição 7 alocada segundo o algoritmo BBR em cada caminho.

agregação entre as reservas. Ao final, a política que seleciona um caminho, o faz no caminho em que a banda residual no intervalo da requisição se tornou mais uniforme. Nesse exemplo, o primeiro caminho foi o selecionado por ter diferença 0 entre a máxima e a mínima banda residual disponível no intervalo da requisição, ao passo que no segundo caminho, tem-se uma diferença de 1 *slot de largura de banda* entre a mínima e a máxima banda residual.

Ambos algoritmos, *Quick-Finish* e Balanceamento de Banda Residual, são utilizados para escalonar aplicações elásticas. Já para escalonar aplicações inelásticas, foi implementado um algoritmo já conhecido da literatura (PATEL; JUE, 2011), como mostra o Algoritmo 4. Assumiu-se que as aplicações inelásticas exigem uma largura de banda fixa durante todo o seu ciclo de vida. Exemplos de aplicações com essas características são aplicações de áudio e vídeo em tempo real, áudio e vídeo armazenado e jogos interativos. Além disso, tais aplicações são alocadas no mesmo instante em que os recursos foram solicitados. A seguir, a descrição das variáveis utilizadas pelo algoritmo:

- **cyclicVar** - Armazena o primeiro caminho a ser analisado pelo algoritmo.
- **path** - Armazena em cada instante uma referência a um dos k menores caminhos.
- **maxBandwidth** - Largura de banda máxima obtida em um dado caminho.

Os dois primeiros parâmetros de entrada, n^s e n^d , referem-se aos nós origem e destino. A duração da requisição está representada pelo intervalo $[t^S, t^E]$. Dentro desse intervalo

Algoritmo 4: Algoritmo para a Reserva Imediata

Entrada: $R = (n^s, n^d, t^S, t^E, bw)$

Saída: aceita ou rejeita a requisição

```

1 cyclicVar ← 0;
2 for path = cyclicVar to path < cyclicVar + k do
3   | for i =  $t^S$  to  $i \leq t^E$  do
4     |   maxBandwidth = MaxBandwidth(path mod k, i);
5     |   if maxBandwidth ≥ bw then
6       |     Aloca previamente bw no path mod k ;
7     |   else
8       |     Flag = TRUE;
9     |     break;
10    |   end
11  |   end
12  |   cyclicVar = cyclicVar + 1;
13 end
14 if Flag == TRUE then
15 |   A requisição nesse path mod k não pode ser aceita;
16 |   Faz um rollback nos recursos alocados nesse caminho.
17 else
18 |   Reserva imediata aceita no path mod k;
19 end

```

a banda exigida é representada pelo bw . Caso em algum instante essa banda exigida não seja satisfeita, a requisição é rejeitada. Diferentemente dos outros algoritmos em que todos os k menores caminhos são analisados para só então tomar uma decisão, aqui, o primeiro caminho a satisfazer a requisição será o utilizado. Para evitar que o caminho 1 dos k menores caminhos acumulasse uma maior quantidade de recursos, criou-se um mecanismo para que a cada nova solicitação um caminho posterior ao último alocado fosse utilizado para iniciar a tentativa de alocação. Para isso a variável *cyclicVar* foi definida como estática, de modo que ela armazena sempre o último caminho em que a alocação ocorreu, não perdendo o seu valor ao sair da função. Alguns comandos contidos no algoritmo fazem parte especificamente da linguagem de programação C, como é o caso da operação $x \bmod y$ que retorna o resto da divisão de x por y .

3.1 ANÁLISE DE COMPLEXIDADE

Esta seção faz uma análise da complexidade computacional do algoritmo Balanceamento de Banda Residual. Como já explicado anteriormente, tanto este algoritmo quanto o *Quick-Finish* fazem uso do algoritmo (YEN, 1971) para o cálculo dos k menores caminhos, no modo *offline*, e por isso não será levado em consideração para melhor visibilidade das diferenças entre o algoritmo proposto e o da literatura.

No algoritmo Balanceamento de Banda Residual, o laço mais externo (linhas 2-15) é o responsável por controlar o número de caminhos e , portanto, é executado k vezes. Dentro desse laço, existe um outro laço responsável por calcular as larguras de banda dentro de todo o intervalo da requisição (linhas 4-6). O intervalo da requisição é representado por $d_{interv} = t^F - t^I$. Para cada iteração neste laço mais interno é executada a função Max-Bandwidth que, como já explicado no algoritmo anterior, possui complexidade $O(|V|)$, sendo V o número de vértices da topologia. Até aqui, a complexidade deste algoritmo portanto é:

- Pior Caso: $O(kd_{interv}|V|)$

O algoritmo BBR difere do QF pelo fato de sempre percorrer todo o intervalo de uma dada requisição para conhecer qual é a capacidade de banda disponível em cada instante. Uma vez conhecida as respectivas capacidades e havendo recursos para aceitar a requisição, percorre-se novamente o intervalo da requisição para fazer a alocação proporcional a disponibilidade da banda (linhas 8-10). O custo para fazer esse procedimento é $O(d_{interv})$. Assim, o custo computacional do algoritmo BBR passa a ser de:

- Pior Caso: $O(kd_{interv}|V| + d_{interv})$

Por fim, as instruções contidas entre as linhas (16-20) não são uma operação dominante do algoritmo, e será executada apenas uma vez ao final do mesmo. Caso exista um ou mais caminhos que satisfaçam a requisição, as linhas (16-18) serão executadas e possuem complexidade $O(k)$ que é o número máximo de caminhos analisados. Assim, a complexidade do algoritmo é:

- Pior Caso: $O(k(d_{interv}|V| + d_{interv}) + k)$

Como se pode perceber, enquanto o BBR sempre precisa percorrer todo o intervalo para então realizar a sua alocação, o QF nem sempre precisa percorrer todo o intervalo. Já no pior caso, o QF realiza um número de operações semelhantes ao BBR. Como foi mostrado na Seção 2.2.1, embora o BBR faça mais operações, ele permite uma maior taxa de aceitação das requisições. Além disso, no dia a dia das redes de circuitos dinâmicos, não é comum se ter uma quantidade muito grande de requisições dentro de um pequeno intervalo de tempo. Desse modo, ambos os algoritmos QF e BBR são compatíveis para atuarem onde foram planejados.

EXPERIMENTOS COMPUTACIONAIS

Para avaliar o desempenho do algoritmo proposto foi desenvolvido um simulador de eventos discretos, implementado através da linguagem de programação C. Nesse experimento, foi utilizado uma simulação baseada em uma topologia real de rede, a NSFNet¹ (do Inglês National Science Foundation Network), como mostra a Figura 4.1. Os *links* são bidirecionais e possuem capacidade de 10 Gbps, abstraídos em 10 *slots de largura de banda* de 1 Gbps cada. Cada nó pode ser a origem ou o destino de um circuito. Para cada requisição, a escolha da origem e do destino segue uma distribuição uniforme. O instante de chegada das requisições segue uma distribuição de *Poisson*, distribuição comumente utilizada em simulações de rede orientada a conexão.

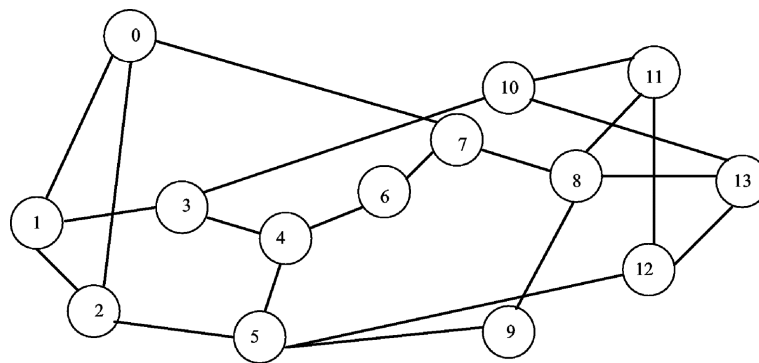


Figura 4.1: *Backbone* NSFNet

Foram utilizados dois tipos de reservas na simulação: Reserva Antecipada (RA) e a Reserva Imediata (RI). Além disso, assumiu-se que toda solicitação de RA é representado por tráfego elástico e toda RI por tráfego inelástico, assim como em (NAIKSATAM; FIGUEIRA, 2007). Essa premissa foi assumida por ser o mapeamento mais comum visto

¹Topologia de rede americana voltada para pesquisa e educação, acessado em http://commons.wikimedia.org/wiki/File:NSFNET_14nodes.svg, na data 20/02/2015

na literatura e para facilitar a visualização dos resultados. Porém, seria perfeitamente possível um tráfego inelástico do tipo streaming querer fazer uma reserva antecipada ou mesmo um tráfego elástico fazer uma reserva imediata.

Em cada simulação foram utilizados um dos dois tipos de algoritmos para escalonar as aplicações elásticas: *Quick-Finish* ou Balanceamento de Banda Residual. Em todas as simulações o algoritmo para alocação de aplicações inelásticas é o mesmo.

Assumiu-se $k=3$ como o número de k menores caminhos. Outros valores foram simulados e, para a topologia já mencionada, os resultados foram similares. O tempo é discretizado em intervalos regulares, de tamanho δ (GUERIN; ORDA, 2000). O período total de simulação foi de 1000 *slots* e todas as requisições que chegam fora da janela de tempo são bloqueadas. Assume-se que o controle da rede é centralizado e a mesma inicia com todos os seus recursos disponíveis.

Além disso, cada experimento foi executado 10 vezes, com diferentes sementes² e o intervalo de confiança para o valor médio foi calculado usando um nível de confiança de 95%. Os recursos são distribuídos uniformemente entre RA e RI.

4.1 RESULTADO PARA DIFERENTES APLICAÇÕES INELÁSTICAS

4.1.1 Probabilidade de Bloqueio Total

Numa tentativa de entender em que circunstâncias o algoritmo proposto é mais apropriado para utilizá-lo, foram concebidos 8 modelos de aplicações inelásticas. Os modelos de 1 à 4 possuem demandas iguais, porém com durações distintas, sendo que o modelo 4 é o que possui maior duração, podendo variar de 1 até 25 *slots* de tempo. Já os modelos de 5 à 8 possuem durações semelhantes, porém com demandas variando entre 10% e 100% da capacidade de um *link* ou caminho na rede. O modelo de aplicação elástica utilizado possui demanda variando entre 1 e 30 *slots de largura de banda*, *book-ahead* variando entre 10 e 25 instantes de tempo e flexibilidade 7. A Tabela 4.1 mostra os quatro modelos utilizados:

Tabela 4.1: Tabela com os parâmetros dos modelos 1 à 4 de RI .

	RA			RI		
	D	α	ω	bw	α	τ
1º Modelo	[1~30]	[10~25]	7	1	0	[1~10]
2º Modelo	[1~30]	[10~25]	7	1	0	[1~15]
3º Modelo	[1~30]	[10~25]	7	1	0	[1~20]
4º Modelo	[1~30]	[10~25]	7	1	0	[1~25]

Os parâmetros D , α , ω , relacionados a reserva antecipada, representam a demanda total de uma aplicação elástica, o seu *book-ahead* e a sua flexibilidade, respectivamente. Os parâmetros bw , α e τ , relacionados a reserva imediata, representam a largura de banda

²No contexto de experimentos científicos, números pseudo-aleatórios são necessários. A produção desses números se baseiam na semente utilizada, de modo que sementes distintas produzem números aleatórios distintos.

de uma aplicação inelástica, o seu *book-ahead* e a sua duração, respectivamente. Nesse primeiro momento, não será o objeto de estudo a flexibilidade das aplicações elásticas e sim os diferentes modelos de aplicações inelásticas. Assim, a flexibilidade adotada para os resultados na Seção 4.1 foi 7.

O primeiro conjunto de resultados mostra como os algoritmos se comportam ao variar a duração das aplicações inelásticas. Na Figura 4.2, tem-se a probabilidade de bloqueio total do sistema ao utilizar o algoritmo *Quick-Finish* e o algoritmo Balanceamento de Banda Residual. A probabilidade de bloqueio total contempla tanto as aplicações inelásticas, quanto as elásticas.

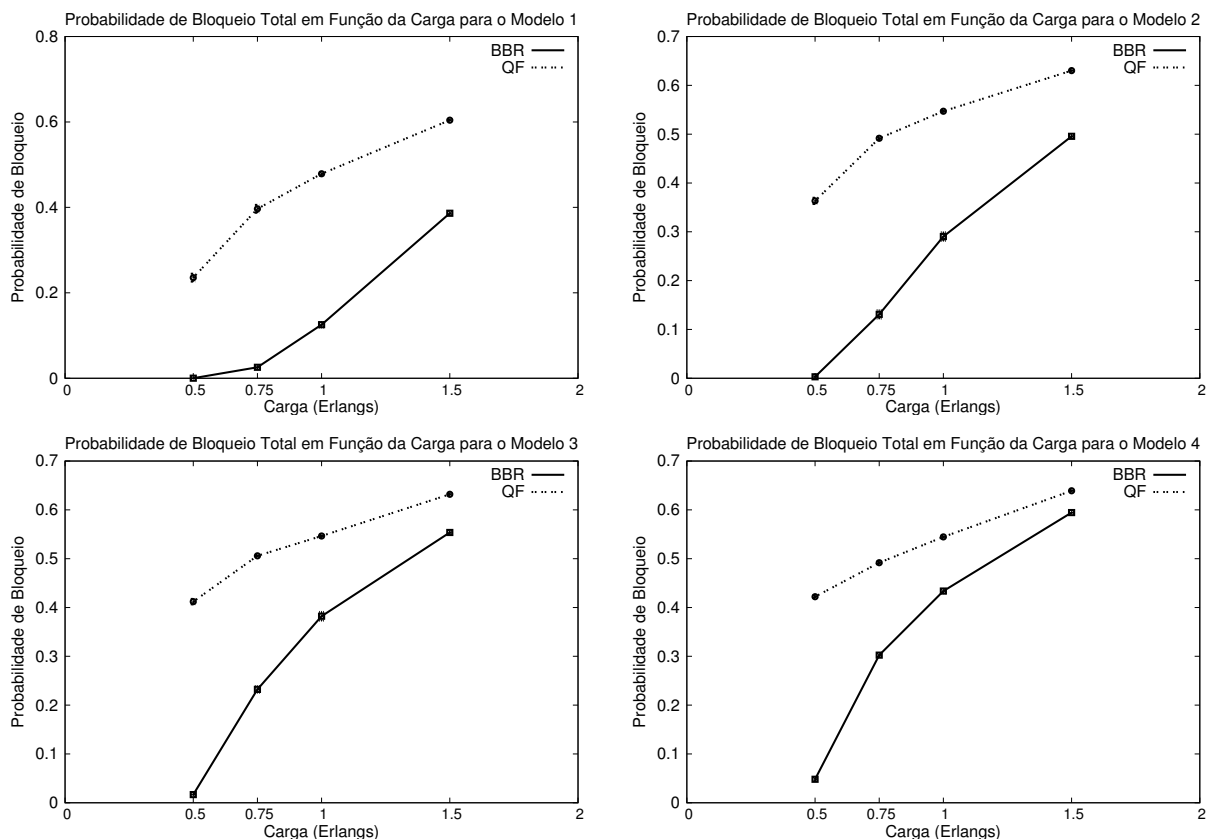


Figura 4.2: Probabilidade de Bloqueio Total em Função da Carga para os Modelos de Aplicação Inelásticas 1 à 4.

Como se pode observar, em todos os quatro primeiros modelos, o algoritmo BBR teve uma menor probabilidade de bloqueio, sendo que as maiores diferenças acontecem para cargas leves e moderadas (0.5 e 0.75 Erlangs). Já para cargas altas (1 Erlang em diante), os dois algoritmos tendem a obter resultados similares.

A explicação é que em situações de alta demanda, os dois algoritmos tendem a preencher seus recursos, de uma maneira ou de outra. Mesmo que um dos algoritmos tenha inicialmente uma vantagem sobre o outro, à medida que mais requisições chegam ambos tendem a conseguir alocar os seus recursos, obtendo assim, ao final da simulação, resultados similares.

Tabela 4.2: Tabela com os parâmetros dos modelos 5 à 8 de RI .

	RA			RI		
	D	α	ω	D	α	τ
5º Modelo	[1~30]	[10~25]	7	[1~2]	0	[1~5]
6º Modelo	[1~30]	[10~25]	7	[1~5]	0	[1~5]
7º Modelo	[1~30]	[10~25]	7	[1~7]	0	[1~5]
8º Modelo	[1~30]	[10~25]	7	[1~10]	0	[1~5]

Já para cargas leves à moderadas, é quando de fato, o modo de alocação dos algoritmos tem um maior impacto no resultado final. Essa é justamente a realidade das redes de circuito dinâmico, como mostra a Figura 4.3.

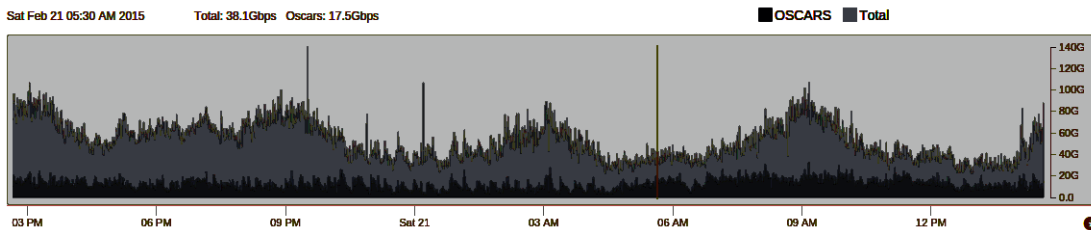


Figura 4.3: Tráfego de dados da rede de circuito dinâmico da ESnet, extraído de <https://my.es.net/>, acessado em 21/02/2015.

Percebe-se que a menor parte do tráfego, representado pela cor cinza mais escura, refere-se ao tráfego submetido via o framework OSCARS para a rede de circuito dinâmico da ESnet.

Portanto, para esse cenário de carga leve, o BBR se mostra com uma menor probabilidade de bloqueio. Essa diferença se acentua ao aumentar a duração de uma aplicação inelástica, pois o algoritmo *Quick-Finish* tem como consequência tornar recursos indisponíveis em determinados instantes de tempo. Assim, quanto maior for a duração da requisição, maiores são as chances de em algum instante dentro intervalo da requisição não ter recursos disponíveis.

Este outro conjunto de resultados mostra o comportamento dos algoritmos para os modelos de aplicação inelástica 5 à 8, no qual ocorre variação na demanda por largura de banda. Os modelos utilizados são mostrados na Tabela 4.2.

Como se pode perceber através da Figura 4.4, à medida que a demanda por largura de banda das aplicações inelásticas aumenta, chegando ao modelo 8 com demanda podendo ser igual a capacidade do *link*, a diferença entre os dois algoritmos diminui. Como pode ser visto no modelo 5, em que o algoritmo BBR consegue os seus melhores resultados comparados aos modelos 6 à 8.

Na Figura 4.4, em apenas um modelo, e para carga alta, o algoritmo BBR acaba tendo um probabilidade de bloqueio um pouco maior. A razão para isso é que o modelo 8 permite que as aplicações inelásticas cheguem a demandar toda a capacidade de um determinado *link* ou caminho, o que faz a maioria dessas requisições serem bloqueadas

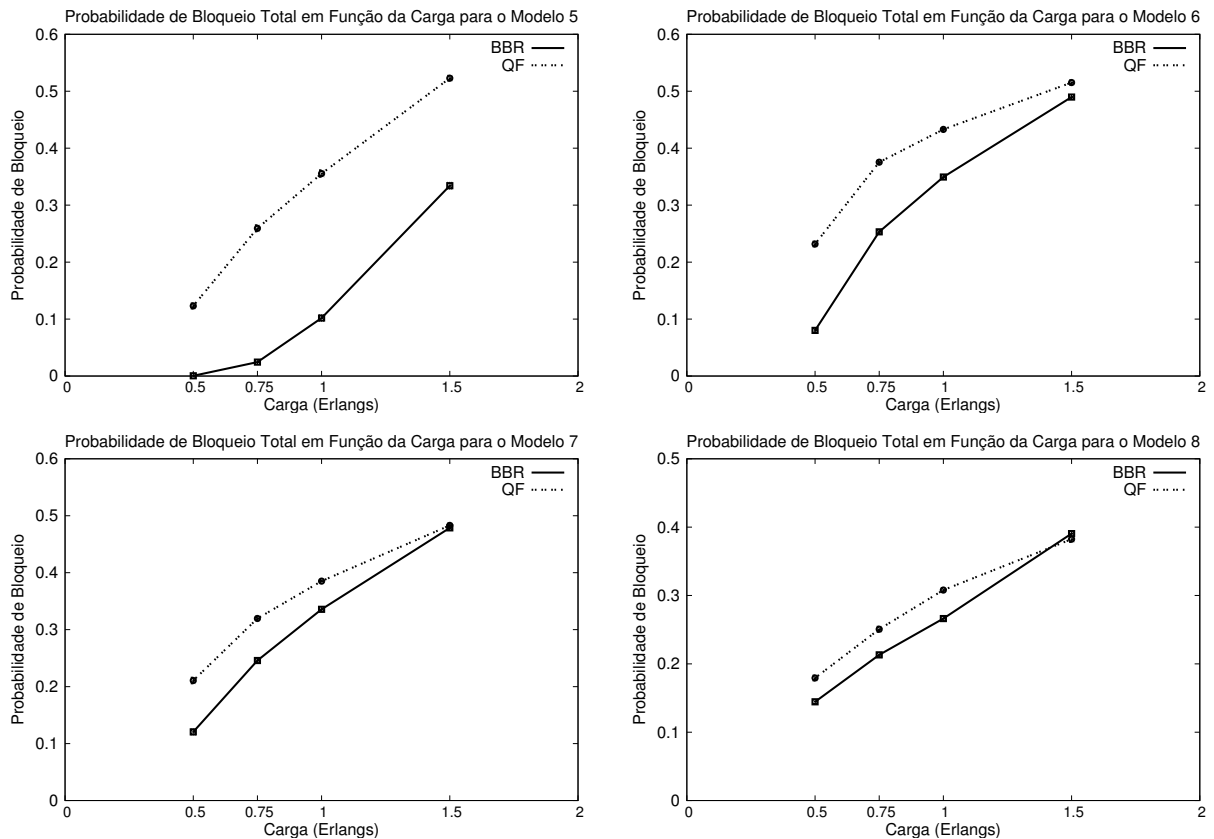


Figura 4.4: Probabilidade de Bloqueio em Função da Carga para Diferentes Modelos de Reserva Imediata 5 à 8.

por ambos algoritmos. Dessa forma, o algoritmo BBR, que foi concebido para conseguir aceitar uma maior quantidade de aplicações inelásticas, acaba não surtindo efeitos.

Como será visto adiante, é justamente na aceitação das aplicação inelásticas que o algoritmo BBR é muito superior em relação ao QF, aumentando assim a diferença entre ambos algoritmos com relação a probabilidade total de bloqueio.

4.1.2 Probabilidade de Bloqueio das Aplicações Inelásticas

Analisando os resultado separadamente para cada tipo de aplicação, a Figura 4.5 mostra a probabilidade de bloqueio apenas para as aplicações inelásticas. A proposta do algoritmo BBR é justamente conseguir aumentar a aceitação desse tipo de aplicação que não se adequa a uma rede com muita variabilidade na disponibilidade dos recursos.

Os resultados, como se pode observar, são bastante expressivos, sendo as maiores diferenças, conforme dito anteriormente, alcançadas para cargas entre 0.50 e 0.75 Erlangs. Com probabilidade média de bloqueio para essas respectivas cargas de 55% e 47% abaixo do algoritmo *Quick-Finish*. A maior diferença foi alcançada para carga 0.50 Erlangs utilizando o modelo 4 que obteve 68% abaixo de probabilidade de bloqueio comparado ao QF. Esses resultados, só não são completamente refletidos na probabilidade de bloqueio total,

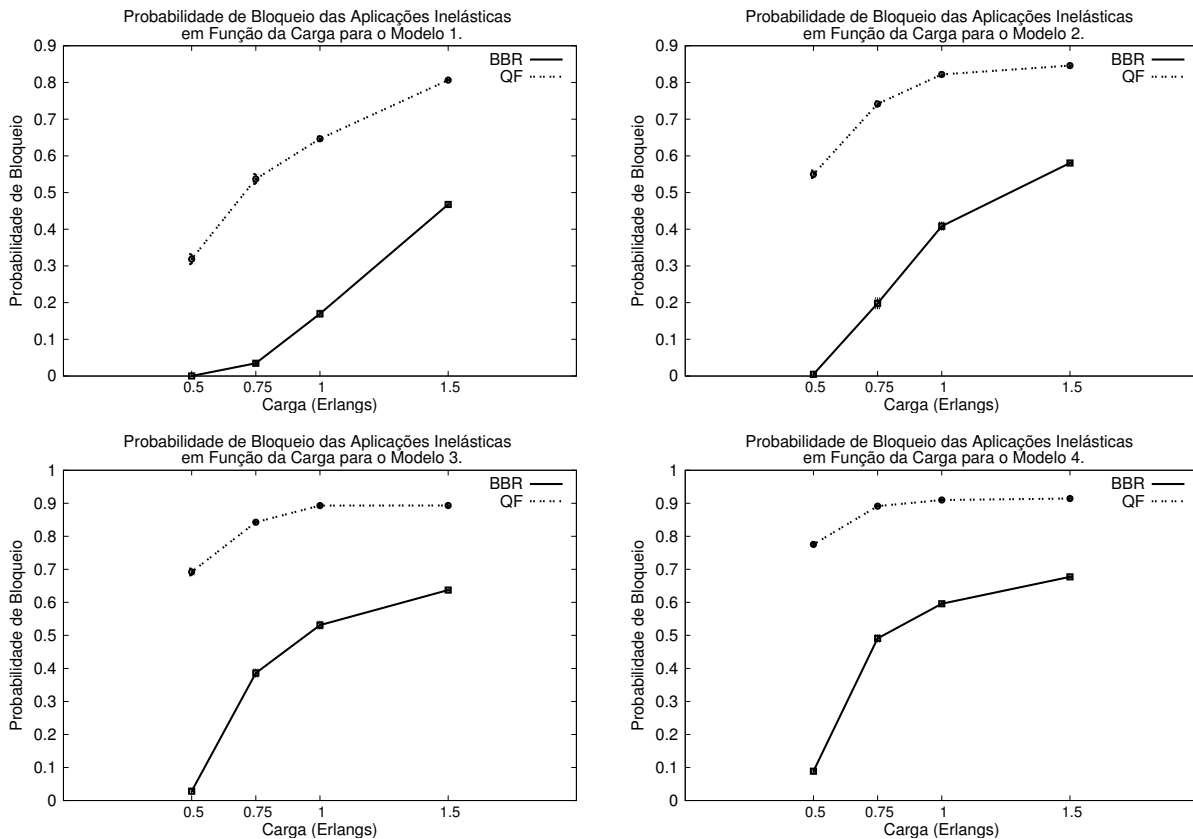


Figura 4.5: Probabilidade de Bloqueio das Aplicações Inelásticas em Função da Carga para os Modelos 1 à 4.

pois o algoritmo Balanceamento de Banda Residual aceita um pouco menos aplicações elásticas comparado ao QF.

Na Figura 4.6, pode-se ver a probabilidade de bloqueio para os modelos 5 à 8. Embora a diferença entre os dois algoritmos diminua um pouco comparado aos modelos de 1 à 4, o BBR continua sendo superior. Dentre os modelos 5 à 8, as maiores diferenças são alcançadas no modelo 5, pois é o modelo com a menor demanda. Conforme já explicado anteriormente, demandas muito altas, acabam não sendo aceitas nem utilizando o BBR nem utilizando o QF.

4.1.3 Probabilidade de Bloqueio das Aplicações Elásticas

Na Figura 4.7, é possível ver a probabilidade de bloqueio das aplicações elásticas ao utilizar os modelos de aplicação inelásticas 1 à 4.

De um modo geral, o algoritmo QF possui uma leve superioridade para cargas altas. Já para cargas leves, os dois algoritmos praticamente possuem os mesmos resultados. Para cargas de 0.50 e 0.75 Erlangs, a diferença entre os dois algoritmos ou é inexistente ou é pequena não chegando a 6%. Para cargas de 1 e 1.5 Erlangs, o algoritmo QF chega a ter uma probabilidade de bloqueio menor que o algoritmo BBR de até 13%.

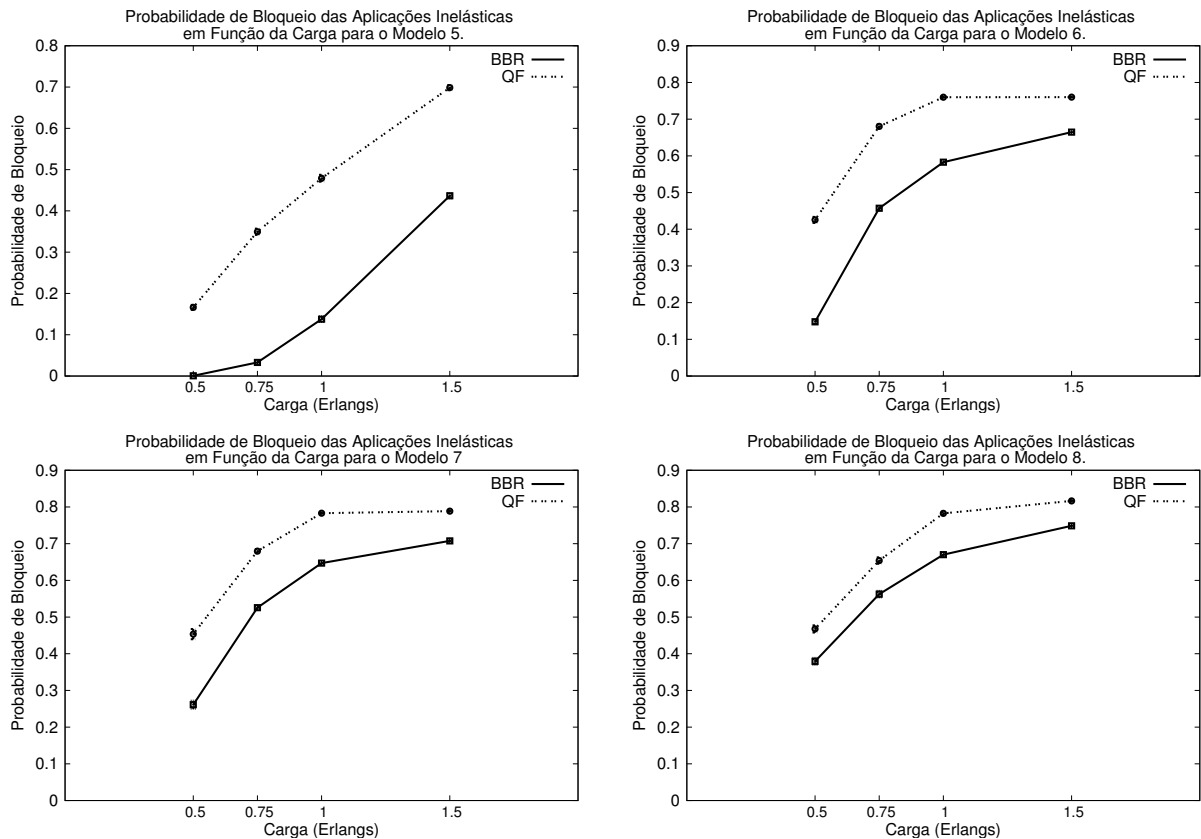


Figura 4.6: Probabilidade de Bloqueio das Reservas Imediatas em Função da Carga para os Modelos 5 à 8.

Essa diferença maior acontece nos modelos que possuem durações mais longas (Modelo 3 e 4). Isso porque quando o algoritmo BBR aceita aplicações inelásticas com durações mais longas, estas por sua vez, acabam consumindo recursos das aplicações elásticas que possuem *book-ahead* curto.

Para os modelos 5 à 8, o algoritmo QF continua a ter uma probabilidade de bloqueio menor em relação ao BBR conforme mostra a Figura 4.8.

Porém, essa diferença é de no máximo 5%. Isso apenas para cargas altas (1 e 1.5 Erlangs), pois para cargas leves os dois algoritmos se comportam de maneira similar. O motivo para essa diferença ter diminuído é que nos modelos 5 à 8 o algoritmo BBR não aceita tantas requisições imediatas comparado ao modelos 1 à 4, uma vez que as exigências são maiores. Dessa forma, como as aplicações inelásticas acabam não sendo aceitas, conseqüentemente irá sobrar recursos para aceitar as aplicações inelásticas que são mais flexíveis.

4.1.4 Taxa de Utilização da Rede

Os resultados a seguir mostram a taxa de utilização da rede. Essa métrica é calculada dividindo a quantidade de recursos consumidos (RC) pelo total de recursos disponíveis

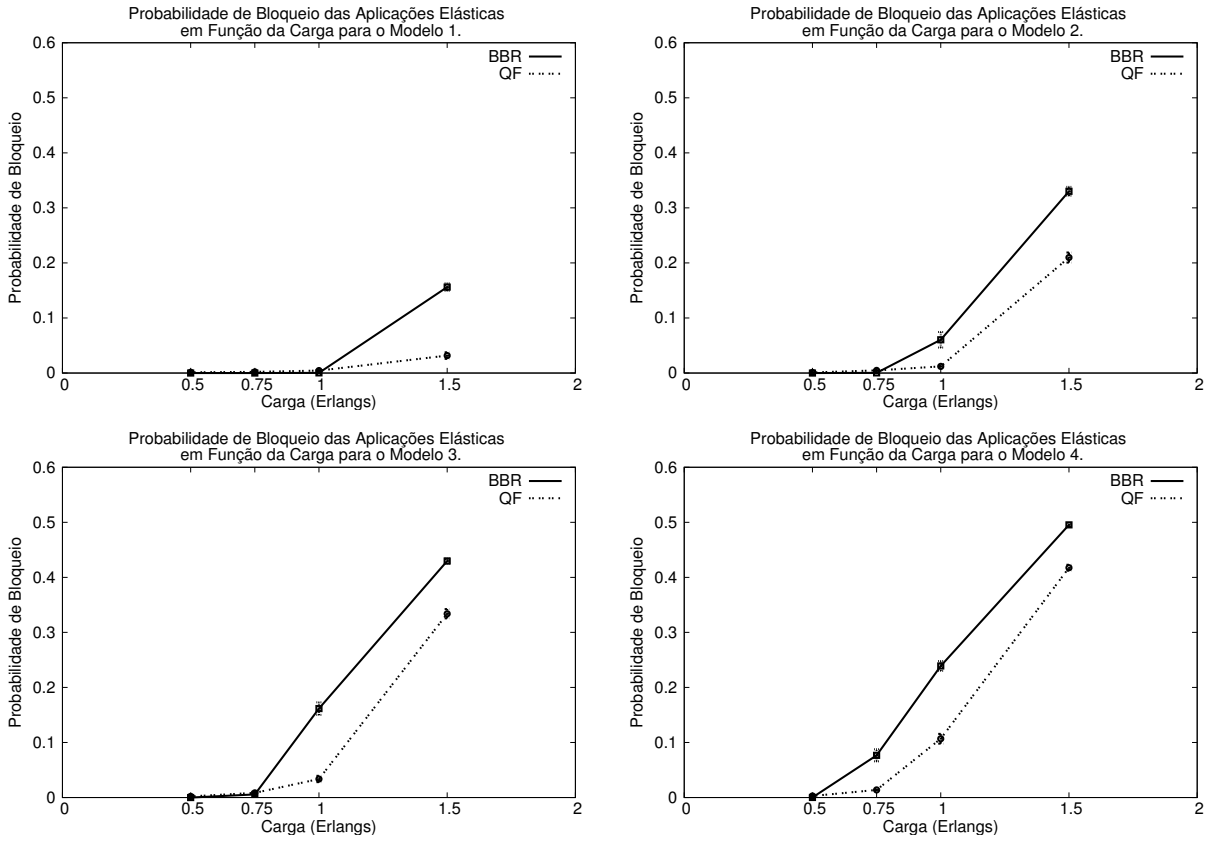


Figura 4.7: Probabilidade de Bloqueio das Reservas Antecipadas em Função da Carga para Diferentes Modelos de Reserva Imediata.

(TR):

$$Util = \frac{RC}{TR} \quad (4.1)$$

O TR é calculado utilizando o número de *links* bem como a sua capacidade e o tempo total da simulação conforme mostra a Equação 4.2 . Nessa simulação específica, a topologia utilizada possui 21 *links* bidirecinais com capacidades de 10 Gbps cada. Assim, a capacidade da rede, por instante de tempo, é de 420 Gbps (21 *links* x 10 slots de banda por *link* x 2 pela bidirecionalidade).

$$TR = NmeroLinks \times CapacidadeLink \times TempoSimulação \quad (4.2)$$

Como o tempo total de simulação foi de 1000 *slots* de tempo, segundo a Equação 4.2, TR vale então 420000. A Figura 4.9 mostra essa taxa de utilização para os modelos de reserva imediata 1 à 4.

Para todos os quatro modelos, a abordagem utilizando o algoritmo Balanceamento de Banda Residual teve uma melhor taxa de utilização. Em todas as cargas e para todos

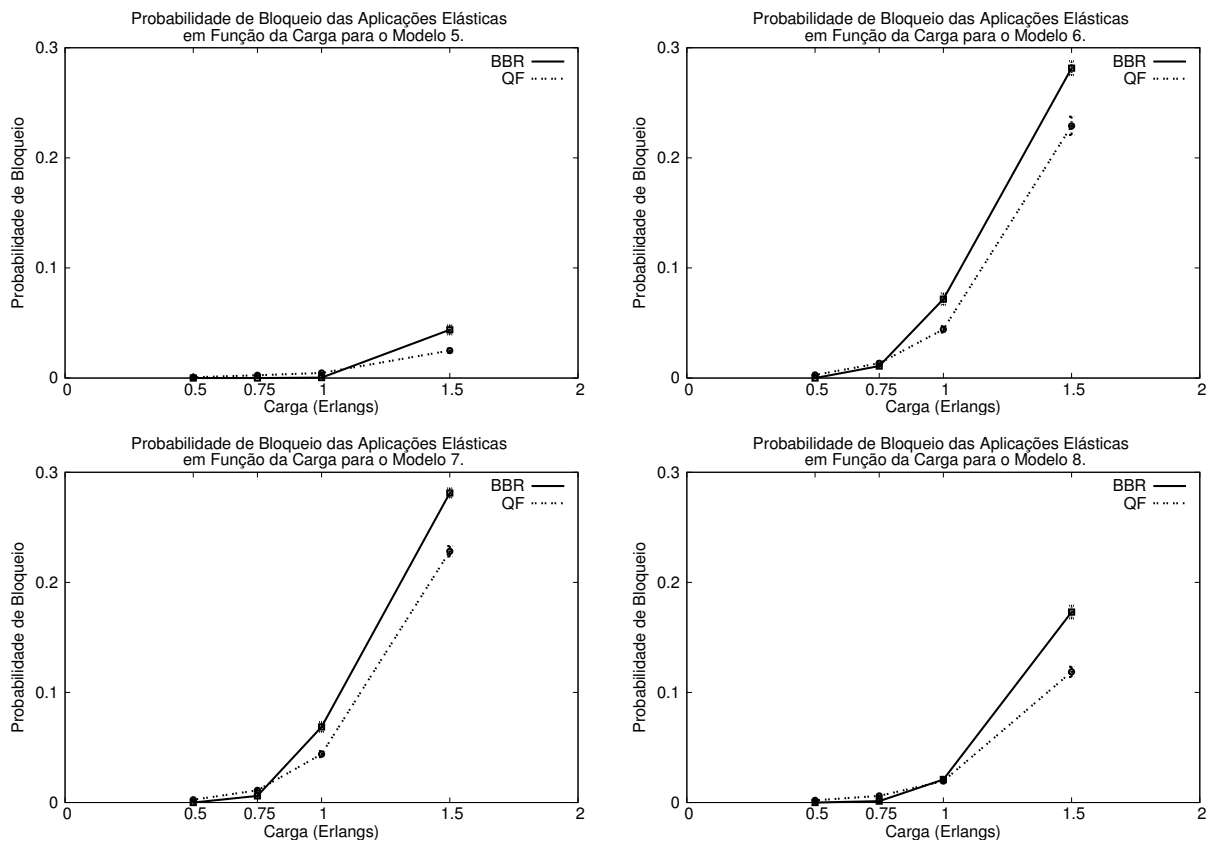


Figura 4.8: Probabilidade de Bloqueio em Função da Carga para Diferentes Modelos de Reserva Imediata.

os modelos de 1 à 4, o algoritmo BBR teve uma utilização melhor de no mínimo 10%. Para o modelo 1, carga de 1 Erlang, o BBR tem uma taxa de utilização de 31% a mais que o algoritmo QF. Nos modelo 2 e 3, a maior diferença acontece na carga 0.75 Erlangs, com 29% e 27% para o modelo 2 e 3, respectivamente. Já o modelo 4, a maior diferença aconteceu na carga de 0.50 Erlangs com 30% a mais de utilização da rede através do algoritmo BBR.

A Figura 4.10 mostra a taxa de utilização para os modelos entre 5 e 8. Para esses modelos a diferença entre o algoritmo BBR e QF é um pouco menor. Ainda assim, os modelos entre 5 e 8 com cargas entre 0.5 e 1 Erlang, o algoritmo BBR obteve uma melhor taxa de utilização.

Em apenas dois casos específicos, modelos 7 e 8 com carga 1.5 Erlangs, o QF teve uma melhor utilização da rede melhor. Embora o algoritmo BBR tenha aceitado um maior número de reservas imediatas, elas não foram suficientes para superar a pequena superioridade entre as reservas antecipadas que o algoritmo QF obteve sobre o BBR.

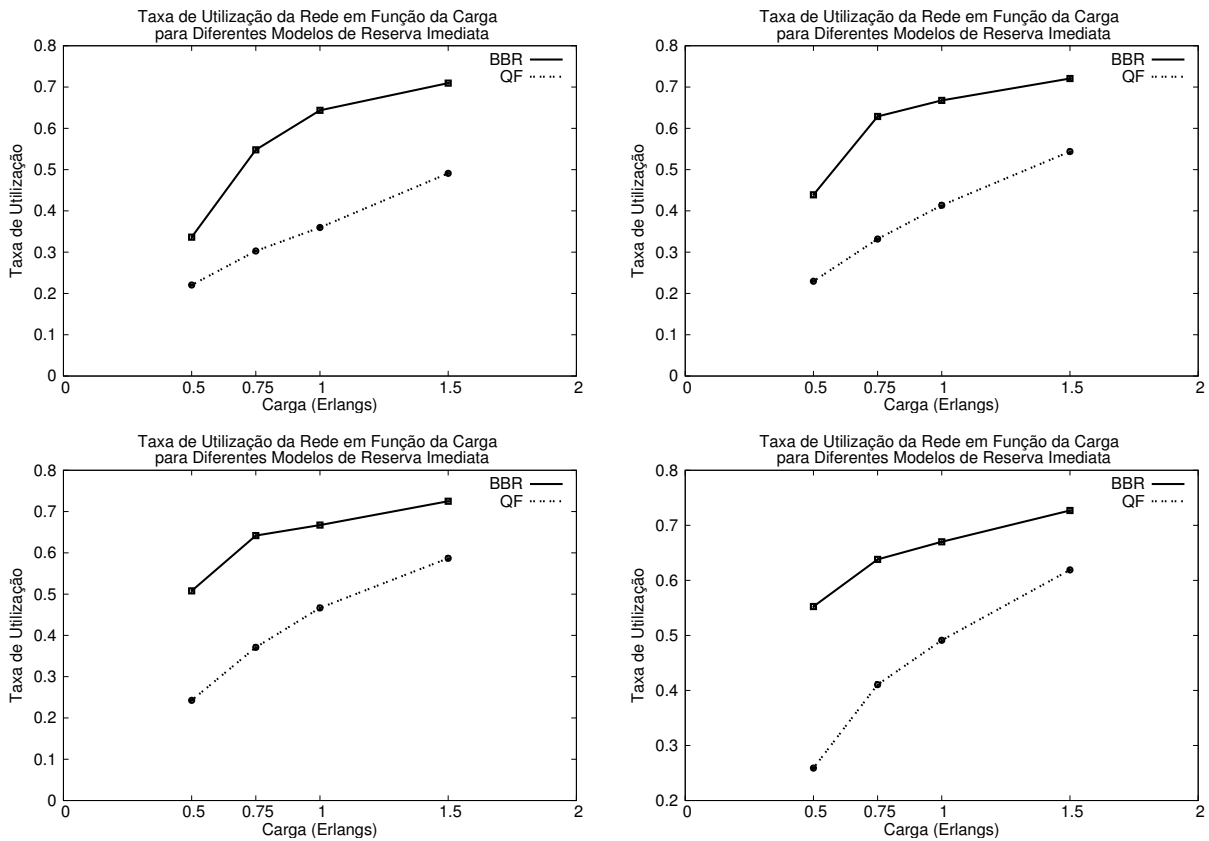


Figura 4.9: Taxa de Utilização da Rede em Função da Carga para os Modelos de Aplicação Inelástica 1 à 4.

4.2 RESULTADOS PARA DIFERENTES FLEXIBILIDADES

Na Seção 4.1, todos os resultados foram obtidos usando apenas um valor para a flexibilidade da requisição, a saber 7. Nesta seção, o intuito é avaliar o impacto que esse parâmetro tem sobre a probabilidade de bloqueio das aplicações elásticas, além também de verificar em que medida esse parâmetro afeta a utilização da rede. Para isso, definiu-se um intervalo de flexibilidade entre 2 e 15. O modelo de aplicação inelástica adotado é o 4 da Tabela 4.1, poderia ser outro.

No primeiro conjunto de resultados, tem-se a probabilidade de bloqueio das aplicações elásticas em função da flexibilidade para quatro cenários de carga, como mostra a Figura 4.11. Quando a carga é de 0.5 Erlangs, não há muita diferença na probabilidade de bloqueio ao variar a flexibilidade, pois a carga oferecida a rede é muito leve. À medida que a carga aumenta, começam a haver diferenças. Por exemplo, no segundo gráfico em que a carga oferecida foi de 0.75 Erlangs, ao variar o valor da flexibilidade de 2 para 5, houve uma melhora significativa para ambos os tipos de algoritmo na aceitação das aplicações elásticas. Porém, essa melhora não evoluiu após novos aumentos de flexibilidade. No caso do QF, após a flexibilidade 5, a probabilidade de aceitação permaneceu quase que constante. Já no BBR, pelo fato de os intervalos de estimativas prováveis terem tido uma

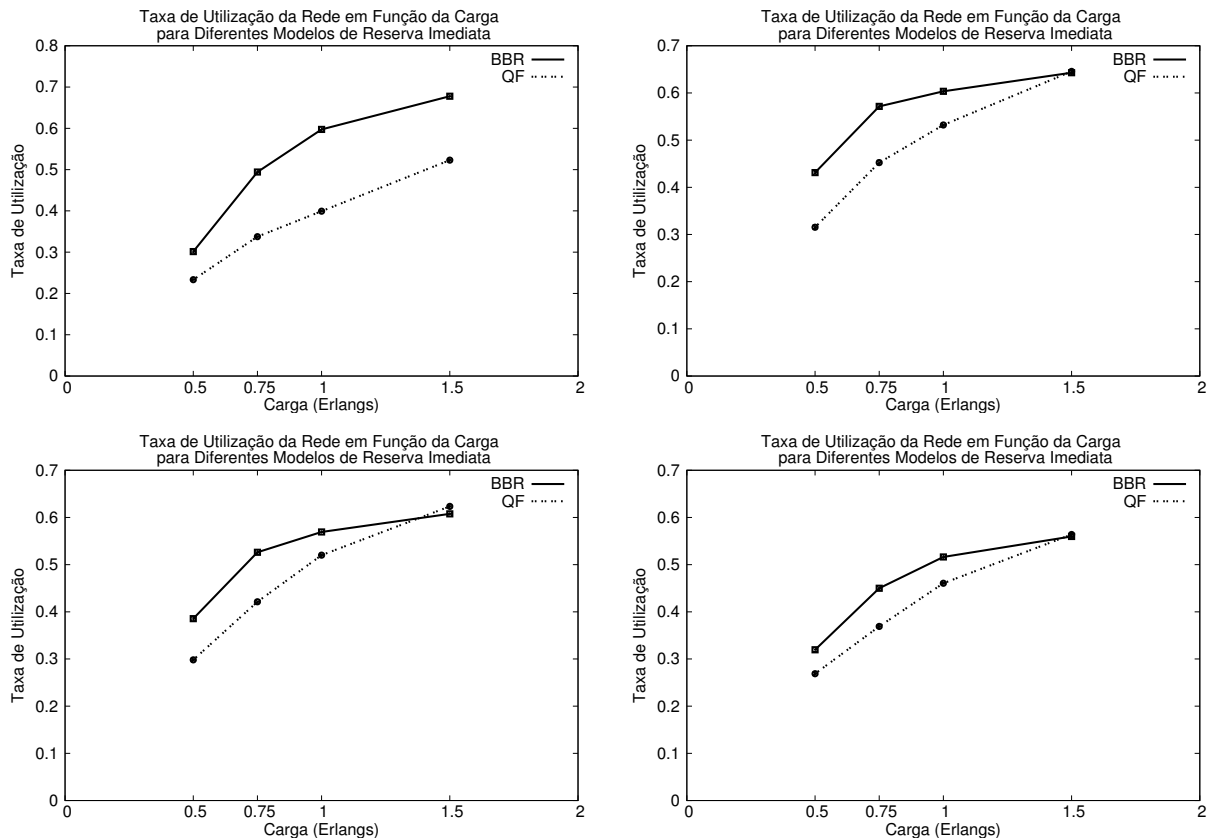


Figura 4.10: Probabilidade de Bloqueio em Função da Carga para os Modelos de Aplicação Inelástica 5 à 8.

oscilação um pouco elevada, nada se pode afirmar com exatidão sobre se houve ou não alteração para o algoritmo BBR.

No terceiro cenário, onde a carga oferecida é de 1.0 Erlangs, à medida que a flexibilidade aumenta os algoritmos BBR e QF começam a ter uma relativa diferença na probabilidade de aceitação das aplicações elásticas. O algoritmo QF se comporta da mesma maneira que no cenário anterior, diminuindo a probabilidade de bloqueio ao saltar da flexibilidade 2 para 5, e nas flexibilidades seguintes se mantendo constante. O algoritmo BBR se comporta de maneira diferente, aumentando a sua probabilidade de bloqueio à medida que a flexibilidade aumenta. A explicação para isso é que pelo fato de o algoritmo BBR explorar toda a extensão da flexibilidade da requisição, conseqüentemente ele acaba adiando a finalização da transmissão cada vez mais para o futuro. Isso gera um certo acúmulo de requisições, o que faz haver mais bloqueio das aplicações elásticas. Por outro lado, essa redução na probabilidade de aceitação faz com que mais recursos estejam disponíveis para requisições imediatas, como será visto no próximo conjunto de resultados. O cenário 4 foi bastante semelhante ao cenário 3, com exceção dos resultados entre as flexibilidades 2 e 5. Neste último cenário, o algoritmo QF já se mantém constante desde o início devido a alta carga, não tendo qualquer efeito a flexibilidade utilizada.

A Figura 4.12 mostra a probabilidade de bloqueio das aplicações inelásticas em função

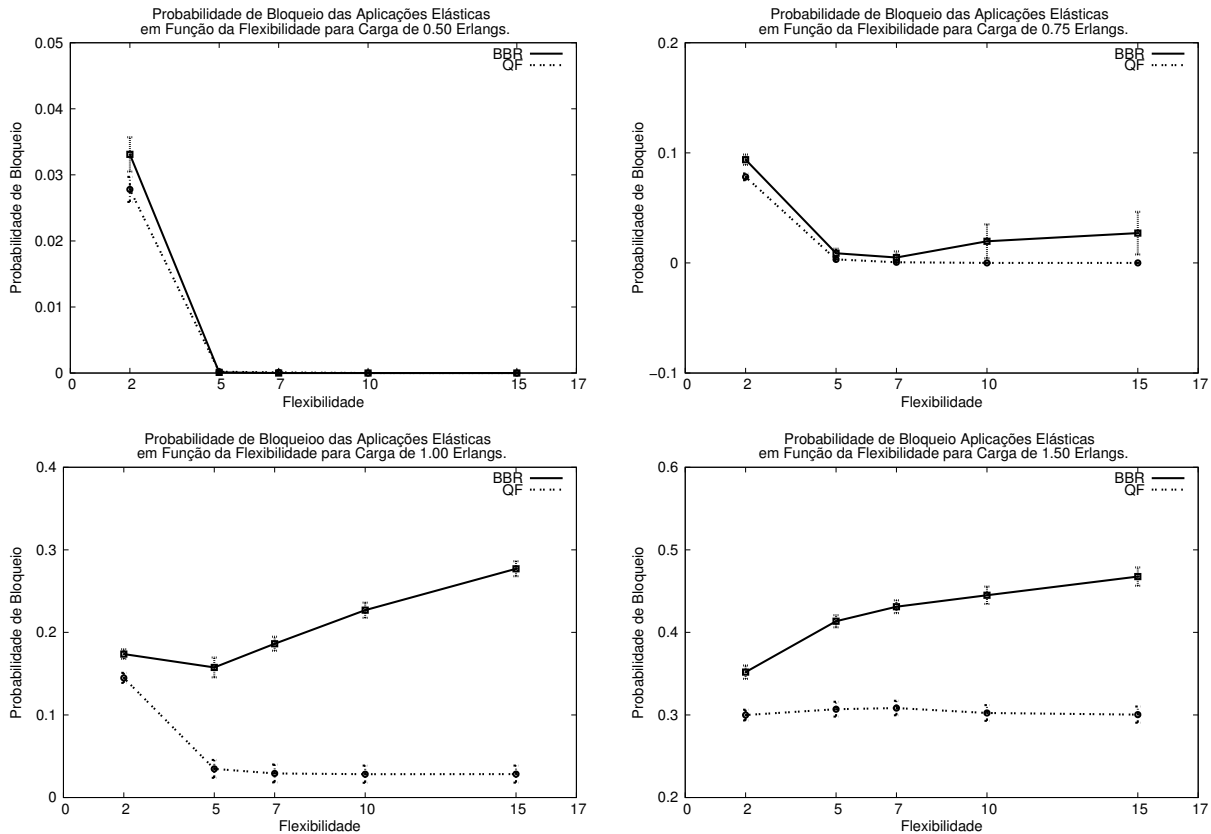


Figura 4.11: Probabilidade de Bloqueio das Aplicações Elásticas em Função da Flexibilidade para Diferentes Cargas.

da flexibilidade para quatro cenários, cada um com uma carga diferente. De um modo geral, o algoritmo QF mantém a sua probabilidade de bloqueio praticamente constante ao variar a flexibilidade para todos os cenários, com exceção do último cenário em que a sua probabilidade acaba variando um pouco.

Por outro lado, ao utilizar o algoritmo BBR, à medida que a flexibilidade aumenta, aumenta-se a probabilidade de aceitação das aplicações inelásticas. Esse crescimento é reflexo das aplicações elásticas que foram bloqueadas, deixando mais recursos disponíveis. A diferença entre os dois algoritmos no que se refere a aceitação das aplicações inelásticas é bastante significativa, tendo reflexo na utilização da rede.

Sob a perspectiva da utilização da rede, a qual computa tanto as aplicações inelásticas quanto as elásticas, a Figura 4.13 mostra os resultados da utilização da rede para diferentes cenários de carga. Em todos os cenários, o algoritmo BBR teve uma maior utilização dos recursos.

No primeiro cenário, por exemplo, o algoritmo BBR, a partir da flexibilidade 5, teve sempre pelo menos o dobro da taxa de utilização alcançada pelo algoritmo QF. No segundo cenário a menor diferença foi de 17% de utilização, no terceiro de 14% e no último cenário a diferença ainda foi de 11%.

Dado o exposto, percebe-se que a introdução da flexibilidade nas requisições tem efei-

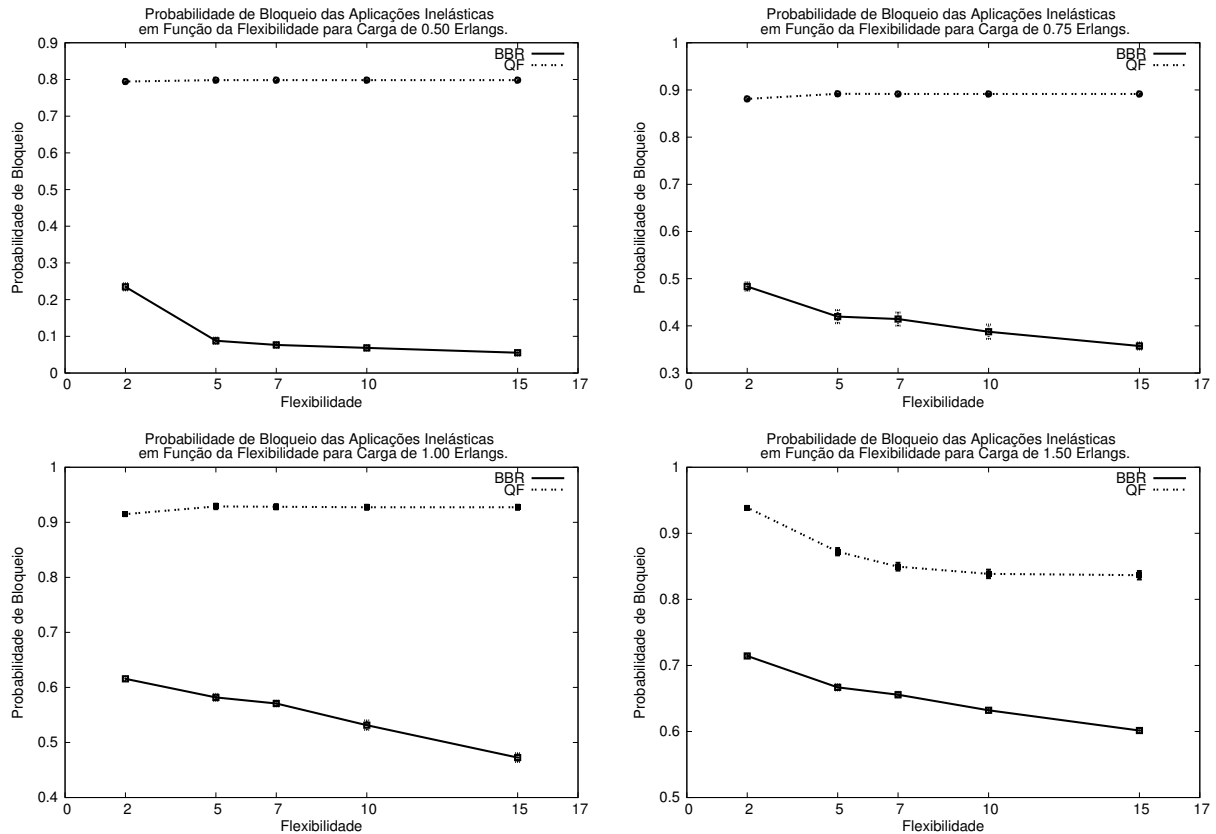


Figura 4.12: Probabilidade de Bloqueio das Aplicações Inelásticas em Função da Flexibilidade para Diferentes Cargas.

tos diferentes a depender da carga da rede e do algoritmo utilizado. No QF, ao aumentar a flexibilidade até um certo ponto, houve uma melhora na probabilidade de bloqueio das aplicações elásticas. Porém, a partir desse ponto a probabilidade se manteve constante, assim como a utilização da rede. Já o algoritmo BBR, ao aumentar a flexibilidade, ele também aumenta a probabilidade de aceitação das aplicações elásticas, porém a partir de uma dada flexibilidade, ele começa a bloquear tais aplicações. Vale ressaltar que esse fenômeno acontece apenas quando a carga da rede está alta.

A partir dos experimentos mostrados neste Capítulo, percebe-se que quando aplicações elásticas são escalonadas obedecendo a regra de conclusão o mais cedo possível (Algoritmo Quick-Finish), isso acaba prejudicando a aceitação de aplicações que não podem ser interrompidas após o início da sua transmissão, como é o caso da maioria das aplicações inelásticas. Esse prejuízo acontece para aplicações que demandam até pouca largura de banda e, se acentua, quando a duração da aplicação aumenta. Além disso, à medida que a carga na rede aumenta, diminui-se a diferença dos efeitos de alocar as aplicações usando o algoritmo *Quick-Finish* ou Balanceamento de Banda Residual. Esse fenômeno é algo esperado, tendo em vista que na medida que várias aplicações tentam alocar recursos, a forma como os algoritmos escalonam tais requisições acabam não tendo tanto impacto no resultado final. Por fim, a utilização de flexibilidade mostrou que para cargas leves e

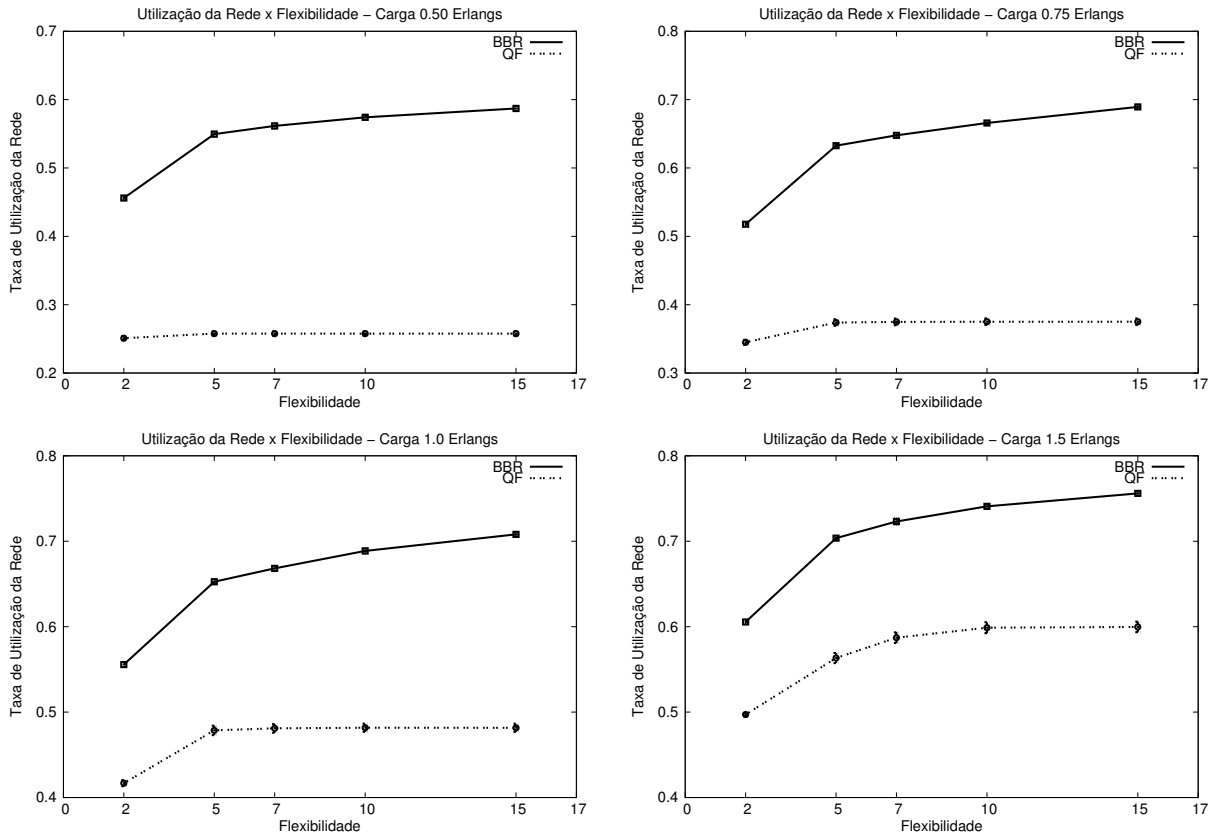


Figura 4.13: Taxa de Utilização da Rede em Função da Flexibilidade para Diferentes Cargas.

moderadas, quanto maior a flexibilidade maiores são os benefícios do BBR em relação ao QF. O mesmo já não acontece para cargas muito elevadas.

Os resultados obtidos neste Capítulo foram feitos utilizando o *Particionamento Estático*. Isso significa que os recursos foram distribuídos de maneira igual entre ambos tipos de reserva, ou seja, metade dos recursos foram destinados para reservas antecipadas e a outra metade para reservas imediatas. Além disso, o número de requisições na rede também foi distribuído igualmente, sendo metade da demanda representado por aplicações elásticas através das reservas antecipadas e a outra metade por aplicações inelásticas através das reservas imediatas. Porém, essas demandas podem variar. É perfeitamente possível haver uma demanda maior de aplicações elásticas em determinados períodos comparado as aplicações inelásticas. Como consequência, o uso do particionamento estático pode ocasionar uma subutilização da rede, uma vez que requisições de um dado tipo podem ser rejeitada mesmo que exista recursos ociosos na outra partição. Portanto, torna-se importante a utilização de uma outra estratégia de gerenciamento de recurso, que será exposta no capítulo seguinte.

GERENCIAMENTO AUTÔNOMO DOS RECURSOS

Os recursos de uma rede são normalmente escassos e possuem um alto custo, por isso devem ser cuidadosamente gerenciados ¹. Existem diversas estratégias de gerenciamento de recursos de rede aplicados nos mais diferentes domínios, como o da telefonia, das redes *wireless* e também das redes cabeadas, voltado para os mais diferentes tipos de aplicações (LAI; MISIC; CHANSON, 1998; HE; WANG; LEIGH, 2006; BORST; MITRA, 2006; MALLA; EL-KADI; TODOROVA, 2001; EL-KADI; OLARIU; ABDEL-WAHAB, 2002). Algumas dessas estratégias podem ou não ser transportadas de um domínio para outro com pequenas adaptações. Esta dissertação implementa uma estratégia para o gerenciamento de recursos, em redes de circuito dinâmico, adaptada da literatura, no qual é permitido fazer uso tanto de reservas antecipadas quanto de imediatas, de maneira justa e com boa utilização da rede.

Na seção de problemas em aberto do *Survey* (CHARBONNEAU; VOKKARANE, 2012), foi colocada uma sugestão para que as reservas imediatas possam de alguma forma ter, pelo menos em algum momento, mais prioridade que as reservas antecipadas. Isso porque, todos os trabalhos que permitem a coexistência das reservas antecipadas com as imediatas e não adotam nenhum tipo de pré-particionamento dos recursos da rede, implicitamente estão dando prioridade para as reservas antecipadas. Isso acontece pois como as reservas antecipadas chegam primeiro, elas irão consumir primeiro os recursos. Assim, em caso de alta demanda de aplicações usando o modelo de reserva antecipada, as aplicações que fizessem uso do modelo de reservas imediatas não teriam chances de encontrar recursos. Esse modelo, sem nenhum pré-particionamento dos recursos, é comumente conhecido na literatura de *Completo Compartilhamento*. Embora ele permita uma maior utilização da rede, a depender do tráfego de entrada, é possível uma degradação significativa na aceitação das reservas imediatas.

Alguns trabalhos que permitem o completo compartilhamento até tentam de alguma forma minimizar o impacto que as reservas antecipadas podem causar sobre as imediatas. Por exemplo, em (ESCALONA et al., 2008), no problema de RWA (Cálculo de Caminho

¹O termo recurso no contexto deste trabalho se refere apenas a largura de banda.

e Atribuição de Comprimento de Onda, do Inglês *Routing and Wavelength Assignment*), uma das estratégias dos autores é escolher um caminho onde exista a maior diferença entre o tempo atual e a primeira reserva antecipada agendada para evitar que elas entrem em conflito por recurso. Isso porque os autores assumem que as reservas imediatas não definem o seu tempo de duração. Contudo, mesmo adotando tal estratégia, em caso de alta demanda por parte das RA, as RI acabam sendo prejudicadas.

Uma forma de solucionar o impacto que as reservas antecipadas têm sobre as reservas imediatas, em casos de alta demanda, seria adicionar um campo de prioridade nas requisições de modo que uma RI possa ter mais alta prioridade que uma RA, como em (XIE; GHANI, 2008). Outra maneira seria realizar algum modo de particionamento, no qual cada tipo de reserva teria recursos exclusivos para si. No contexto do problema de RWA, os autores em (HE; WANG; LEIGH, 2006), por exemplo, fazem uso do particionamento estático, atribuindo as reservas antecipadas cinco, dos oito primeiros comprimentos de onda, e o restante para as reservas imediatas. Desse modo, mesmo havendo alta demanda de RA, as RI teriam, independentemente da carga, os seus recursos. Essa foi a forma de alocação utilizada no capítulo anterior denominada de *Particionamento Estático*.

Em (ROUSSEAU; TRIAY; VOKKARANE, 2013), os autores definem previamente uma quantidade fixa de recursos, distribuídos igualmente para cada classe de reserva, sendo denominado de recurso originário da classe. Por outro lado, um recurso não originário é aquele recurso que não foi destinado inicialmente para uma dada classe. Para fazer um bom uso dos recursos, os autores permitem que requisições de uma dada classe possam utilizar recursos de uma outra classe, desde que sejam passíveis de serem interrompidas (preemptadas) em determinados casos. Ou seja, quando o limite de recurso originário de uma classe de reserva for alcançado, permite-se utilizar o recurso de uma outra classe, desde que a requisição alocada fora do seu recurso originário possa ser preemptada por reservas que demandam nos seus recursos originários.

Nesta dissertação, também é defendido um esquema de particionamento como forma de minimizar o impacto que as reservas antecipadas causam sobre as imediatas. Portanto, foi proposto uma estratégia de gerenciamento de recursos modificada de (ROUSSEAU; TRIAY; VOKKARANE, 2013), que funciona da seguinte maneira: ao iniciar a rede, atribui-se uma quantidade de recurso para cada classe de reserva, sendo que essa atribuição só vale apenas para um determinado intervalo de tempo e não por toda a duração da simulação. Isso porque, de tempos em tempos, novos particionamentos de recursos são feitos dinamicamente para atender a demanda de tráfego corrente. Além disso, dentro de cada intervalo, o gerenciador poderá alocar os recursos destinados para uma determinada classe de reserva da maneira que for mais apropriada. Ou seja, o gerenciador do recurso poderá deixar que uma classe de reserva consuma todo o seu recurso, por exemplo, no início, no fim, ou mesmo de forma distribuída, desde que compense nos outros instantes, como mostra a Figura 5.1. Isso é parte da estratégia proposta e está referenciado neste trabalho de *Particionamento Dinâmico*.

Uma boa opção para se definir a quantidade de recursos destinados a reserva antecipada seria se basear na demanda das reservas imediatas. Porém, como não se sabe antecipadamente qual será essa demanda, uma boa estratégia é levar em consideração o histórico de demanda das reservas imediatas. Assim, foram criados pontos de coleta

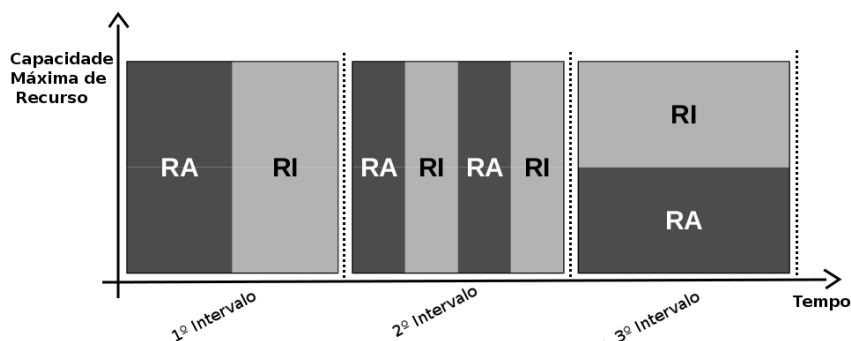


Figura 5.1: Diferentes Modos de Alocação de Recursos dentro de um Intervalo.

de tráfego, em intervalos regulares de 100 instantes de tempo, para medir a demanda de tráfego de cada classe como mostra a Figura 5.2. Definiu-se no total 10 pontos de coleta. No contexto da simulação realizada, esse é um número razoável em que se consegue ter uma boa quantidade de demandas para reservas antecipadas e imediatas. Porém, poderia-se também utilizar uma outra forma para definir os instantes de ponto de coleta, por exemplo, baseado em algum evento.

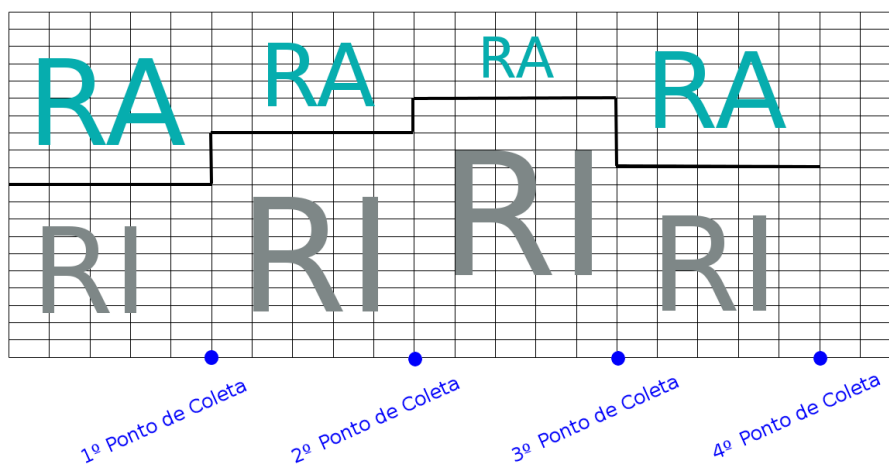


Figura 5.2: Distribuição dos Recursos entre Reserva Antecipada (RA) e Reserva Imediata (RI) em Função da Demanda.

O cálculo realizado em cada ponto de coleta é feito utilizando a média móvel ponderada, selecionando os últimos três pontos de coleta, de modo que o histórico de consumo do último intervalo tem sempre o maior peso, pois reflete a tendência atual na demanda dos recursos. Com base nesse histórico, é que se projeta a distribuição de recurso para o próximo intervalo. Por exemplo, se em um determinado intervalo, a distribuição de recurso for igualmente distribuída entre as classes, sendo que uma delas não utilizou todo o seu recurso, enquanto a outra não só utilizou todo o seu recurso como bloqueiou requi-

sições por falta dele. Nesse caso, haverá uma distribuição maior de recursos no próximo intervalo para aquela classe em que houve uma maior demanda.

Foi adicionada a estratégia de gerenciamento de recurso denominada de *Particionamento Dinâmico* à capacidade de interromper requisições uma vez que elas tenham ultrapassado o limite de recursos definidos previamente. A essa nova estratégia, que é a proposta desta dissertação, dar-se o nome de *Particionamento Dinâmico com Preempção*. Desse modo, as reservas antecipadas que ultrapassarem o seu limite definido para um dado intervalo do tempo, serão passíveis de serem interrompidas. Esse é um mecanismo para prevenir uma possível mudança repentina na demanda de tráfego.

O funcionamento dessa proposta é apresentado através do Pseudo-Algoritmo 5. Vale ressaltar, que foi colocado um resumo da parte mais importante para entender como o mesmo funciona. Na prática, o algoritmo é um pouco mais complexo.

O Algoritmo 5 recebe como entrada uma requisição e decidirá se esta será ou não aceita no sistema. Inicialmente, como não se sabe nada sobre a demanda que irá chegar, os recursos são distribuídos igualmente apenas até o primeiro ponto de coleta (Linha 1 do Algoritmo 5). A partir do primeiro ponto de coleta em diante, toda a distribuição é baseada no histórico da rede.

Para aceitar uma reserva antecipada, primeiro é necessário verificar se existe recurso destinado para RA disponível. Caso exista, a reserva é aceita com garantia total de que a mesma concluirá no tempo alocado (Linhas 19-21). Caso contrário, o algoritmo permite que uma RA possa utilizar os recursos destinados para RI (Linhas 22-24), desde que possam ser interrompidas caso uma RI venha necessitar utilizar o seu recurso. Essa é uma estratégia de minimizar uma possível mudança repentina do tráfego. Por exemplo, o histórico previu uma determinada demanda de tráfego que não se concretizou. Justamente para esse cenário que a ação de fazer uso dos recursos de outra classe pode trazer um certo benefício para a rede. Caso não exista nem recursos disponíveis para RA nem para RI, a requisição é bloqueada (Linhas 25-27).

Já para aceitar uma requisição que solicite uma reserva imediata, basta inicialmente verificar se há recurso destinados para RI que permitam aceitar a demanda da requisição (Linha 8). Caso exista, a requisição é aceita normalmente (Linha 9). Senão, o gerenciador do recurso verifica se alguma reserva antecipada usou recursos destinados a reserva imediata. Caso isso seja um fato, tal requisição se encontrará numa lista de requisições que podem ser interrompidas (Linha 11). Se isso se confirmar, uma ou mais reservas antecipadas que coincidirem os instantes de tempo com a reserva imediata serão interrompidas e a requisição corrente é resubmetida ao sistema (Linha 12). Caso ainda assim, a requisição não consiga ser aceita, a mesma é bloqueada (Linha 15).

Algoritmo 5: Particionamento Dinâmico com Preempção

Entrada: *Req***Saída:** aceita ou rejeita a requisição

```

1 Inicialmente os recursos são distribuídos de forma igual entre RA e RI, até o
  primeiro ponto de coleta;
2 while (tempoCorrente < tempoSimulação) do
3   if Chegou em um próximo ponto de coleta then
4     | Verifica qual foi a demanda de RA e RI e calcula a distribuição dos recursos
      com base nessa demanda;
5     | Atualiza as variáveis que controlam esses recursos com base nesse cálculo;
6   end
7   if (Req == ReservaIMEDIATA) then
8     | if Demanda de Req for menor que recursos destinados para RI then
9       | Aceita a requisição e atualiza os recursos restantes para RI;
10    end
11    else if (Se houver requisições preemptáveis) then
12      | Remove tais requisições e tenta aceitar a requisição novamente.;
13    end
14    else
15      | Rejeita a requisição e registra tal demanda que não foi atendida para
        planejamento futuro da distribuição dos recursos;
16    end
17  end
18  else if (Req == ReservaANTECIPADA) then
19    | if Demanda de Req for menor que recursos destinados para RA then
20      | Aceita a requisição e atualiza os recursos restantes para RA;
21    end
22    else if (Houver recursos disponíveis destinados para RI) then
23      | Aloca a requisição e a coloca em uma lista de requisições passíveis de
        serem interrompidas. Essa requisição não terá garantia se a sua reserva
        irá se concluída no tempo alocado.
24    end
25    else
26      | Rejeita a requisição e registra tal demanda que não foi atendida para o
        planejamento futuro da distribuição dos recursos;
27    end
28  end
29 end

```

5.1 EXPERIMENTOS COMPUTACIONAIS

Como dito anteriormente, em um cenário real, será muito comum que existam diferentes aplicações demandando reservas antecipadas e imediatas no decorrer do tempo. O intuito deste experimento é justamente verificar o impacto que os diferentes modelos de gerenciamento de recurso causam na utilização da rede e na probabilidade de aceitação das requisições. A estratégia proposta é independente do algoritmo de alocação de largura de banda. Por esse motivo, apenas um algoritmo foi escolhido, o BBR.

O modelo de tráfego estático e proporcional é bastante simples, porém não muito realístico. Esse foi o modelo de tráfego usado durante a simulação da Seção 4.1, pois naquele momento o objetivo maior era dar enfoque aos algoritmos de alocação de largura de banda. Agora, faz-se necessário repetir o mesmo tipo de tráfego, porém com o objetivo de verificar o comportamento dos modelos de gerenciamento de recurso. Além disso, outros modelos de tráfego como o tráfego dinâmico e também desproporcional são criados para realizar a análise. Ao todo, definiu-se 8 cenários de demandas distintas por reservas antecipadas e imediatas, conforme mostra a Tabela 5.1:

Tabela 5.1: Tabela com os diferentes cenários de tráfego.

Cenário	Descrição
1	Baixa demanda de RA e RI
2	Alta demanda de RA e RI
3	Alta demanda de RA, baixa demanda de RI
4	Alta demanda de RI, baixa demanda de RA
5	Demanda crescente de RI, demanda decrescente de RA
6	Demanda crescente de RA, demanda decrescente de RI
7	Alta demanda de RI, baixa demanda RA com inversões repentinas
8	Alta demanda de RA, baixa demanda RI com inversões repentinas

A especificação exata do que vem a ser alta ou baixa demanda está representado na Tabela 5.2. Nos cenários de 1 à 4 o tráfego é fixo, ou seja, a mesma carga é oferecida durante toda a simulação. Nos cenários 5 à 8, a carga oferecida muda ao longo do tempo. Nos cenários 5 e 6, a demanda crescente é uma carga que se inicia baixa e ao longo do tempo vai crescendo. A demanda decrescente é o oposto da demanda crescente. Já os cenários 7 e 8 são semelhantes aos cenários 3 e 4, sendo a única diferença a questão da inversão de demanda. A inversão nada mais é do que a troca do padrão de demanda oferecido a rede por um curto instante de tempo, ou seja, se existe alta demanda de RA e baixa de RI, a inversão inverte esse padrão fazendo com que exista baixa demanda de RA e alta de RI por um pequeno instante de tempo, e em seguida, volta-se ao padrão comum oferecido. Assumiu-se que nos cenários de tráfego em que há mudanças na carga oferecida, essas são feitas nos mesmos instantes em que ocorrem as medições, ou seja, nos pontos de coleta de tráfego. Além disso, quando se diz que a demanda por um dado tipo de reserva alterou 3 vezes, as mudanças foram feitas distribuídas ao longo da simulação.

Além dos diferentes cenários de carga, foram definidos diferentes estratégias de gerenciamento de recursos: Particionamento Estático (Estratégia 1), Completo Comparti-

Tabela 5.2: Tabela com as cargas para diferentes demandas de RA e RI..

Cenário	Carga Total (Erlangs)	Carga para RA	Carga para RI
1	0.50	0.25	0.25
2	2.00	1.00	1.00
3	2.00	1.75	0.25
4	2.00	0.25	1.75
5	2.00	1.75, 1.00, 0.25	0.25, 1.00, 1.75
6	2.00	0.25, 1.00, 1.75	1.75, 1.00, 0.25
7	2.00	0.25, 1.75, 0.25	1.75, 0.25, 1.75
8	2.00	1.75, 0.25, 1.75	0.25, 1.75, 0.25

lhamento (Estratégia 2), Particionamento Dinâmico (Estratégia 3) e o Particionamento Dinâmico com Preempção (Estratégia 4).

O primeiro conjunto de resultados mostra a probabilidade de aceitação das reservas antecipadas para os diferentes cenários de tráfego, segundo as quatro estratégias de gerenciamento de recurso. O resultado pode ser visto na Figura 5.3. Como se pode perceber, em todos os cenários a estratégia 2, completo compartilhamento, teve a melhor taxa de aceitação. Esse resultado era esperado, uma vez que a única limitação para bloquear uma dada RA é quando de fato não existe mais recursos para as exigências da requisição corrente. Como consequência, a estratégia 2 tem o pior desempenho com relação a taxa de aceitação das RI como será mostrado a seguir. No cenário 1, devido a carga ser pequena, todas as estratégias tiveram os mesmos resultados, aceitando todas as requisições. No cenário 2, em que existe alta carga para ambos tipos de requisição, a estratégia de completo compartilhamento obteve novamente as melhores taxas de aceitação. No cenário 3, devido a existência de alta demanda de RA, muitas delas acabam sendo bloqueadas e, por isso, a taxa de aceitação é menor comparada ao cenário 1. Ainda no cenário 3, o particionamento estático teve o pior desempenho uma vez que ele não é regulado baseado na demanda e sim em um particionamento feito no início do funcionamento da rede. As outras estratégias, por se basear na demanda, destinaram mais recursos para RA e por isso tiveram uma aceitação melhor. No cenário 4, como houve pouca demanda de RA, as estratégias 1 e 2 de fato tem a melhor taxa de aceitação, uma vez que o primeira já reserva metade dos recursos para RA independente da carga e a segunda estratégia sempre aceita RA enquanto há recursos.

No cenário 5 e 6, novamente a estratégia de completo compartilhamento teve a maior probabilidade de aceitação para reservas antecipadas e o estático compartilhamento a menor probabilidade. No particionamento dinâmico e no particionamento dinâmico com preempção, os resultados foram bastante próximos com uma aceitação um pouco maior por parte da última estratégia. Os cenários 7 e 8 foram projetados para verificar em que circunstâncias as estratégias eram capaz de se adaptar a possíveis alterações bruscas na tendência de tráfego. A estratégia 4 tenta minimizar justamente essa questão, tomando ações pró-ativas evitando que possíveis má projeções nas tendências da demanda possam prejudicar o planejamento do uso dos recursos. Assim, a estratégia 4 ocupou o segundo lugar, perdendo apenas para a estratégia de completo compartilhamento.

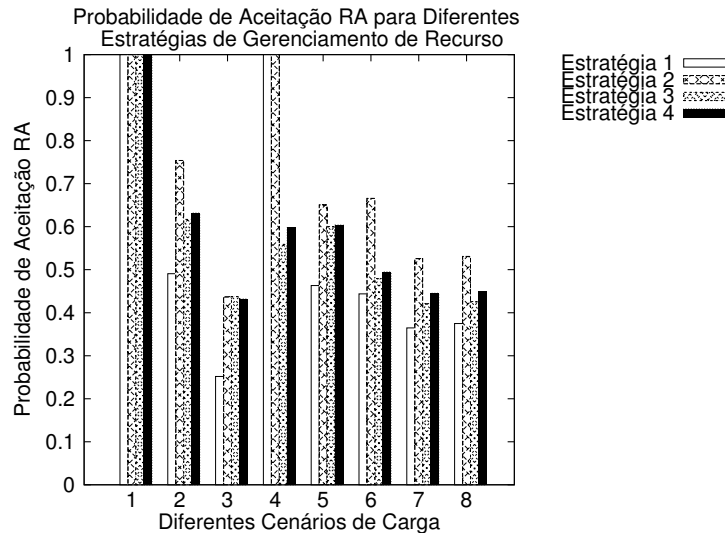


Figura 5.3: Probabilidade de aceitação das reservas antecipadas para diferentes cenários utilizando diferentes estratégias de gerenciamento dos recursos.

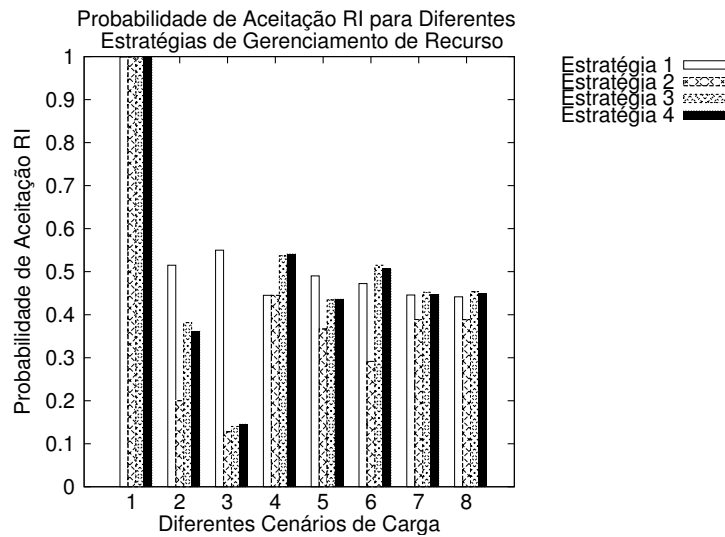


Figura 5.4: Probabilidade de aceitação das reservas imediatas para diferentes cenários utilizando diferentes estratégias de gerenciamento dos recursos.

A Figura 5.4 mostra a probabilidade de aceitação das requisições imediatas. Novamente, como no cenário 1 existe pouca demanda as estratégias não se diferenciam uma da outra, porém no cenário 2 é possível perceber que o pior desempenho é alcançado pela estratégia de completo compartilhamento. Aqui, vale destacar a diferença na probabilidade de aceitação para os dois tipos de reservas que essa estratégia promove, sempre beneficiando as reservas antecipadas. No cenário 3, devido o particionamento estático não considerar a demanda das requisições e deixar metade dos recursos disponíveis para as reservas imediatas, em caso de pouco demanda por parte das RI, ele acaba tendo a melhor probabilidade de aceitação. As outras estratégias acabam disponibilizando mais recursos para as reservas antecipadas devida a sua alta demanda, o que faz com que exista uma pequena quantidade de recursos destinadas as reservas imediatas, promovendo, assim, uma maior quantidade de bloqueio.

O cenário 4 é mais um cenário em que se percebe a diferença na probabilidade de aceitação entre as reservas antecipadas e imediatas causadas pela estratégia do completo compartilhamento. Aqui, as estratégias 3 e 4 tiveram um melhor resultado e conseguiram manter de certa forma uma nível de probabilidade semelhante ao conseguido nas reservas antecipadas.

Nos cenários 5, 6, 7 e 8, esse comportamento se repete, tendo sempre a estratégia 2 com o pior desempenho. Ou seja, a estratégia 2 teve o melhor de desempenho na probabilidade de aceitação das reservas antecipadas, contudo tem o pior desempenho na probabilidade de aceitação das reservas imediatas.

Por fim, analisou-se a utilização da rede obtida pelas diferentes estratégias de gerenciamento dos recursos, como mostra a Figura 5.5. O cenário 1 é onde existe a menor utilização da rede pois a carga oferecida foi baixa. No cenários seguintes, a utilização teve uma média de 70 a 80% de utilização. Além disso, é facilmente percebido que nos dois extremos encontram-se a estratégia 1 e 2. A estratégia 1 tem o pior desempenho na utilização e a estratégia 2 o melhor desempenho. No meio termo, encontram-se as estratégia 3 e 4. Como já dito anteriormente, a estratégia 2 alcança esse desempenho bloqueando mais reservas imediatas que antecipadas. Por outro lado, nas estratégias 3 e 4 não existe essa diferença tão grande.

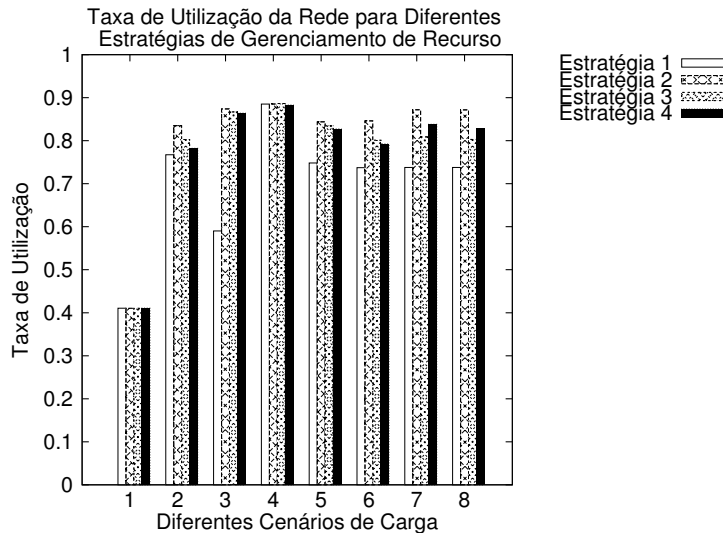


Figura 5.5: Taxa de utilização da rede para diferentes cenários utilizando diferentes estratégias de gerenciamento dos recursos.

Nos cenários em que há variações bruscas nas demandas, cenários 7 e 8, pode-se perceber que a estratégia 4 se comporta melhor comparada a estratégia 3, o que resulta em uma melhor utilização da rede. Isso acontece, porque na estratégia 4 é permitido que se aceite mais reservas antecipadas do que o seu limite estipulado, contanto que possam ser interrompidas. Essa ação faz com que em casos de haver uma projeção equivocada de RI, a estratégia 4 ainda assim faça um bom uso dos recursos.

A proposta de gerenciamento de recurso apresentada aqui teve o seu propósito atendido no sentido de alcançar uma boa utilização da rede e ao mesmo tempo diminuir as diferenças entre reservas antecipadas e imediatas no que diz respeito a probabilidade de aceitação. Embora a estratégia 2 tenha obtido os melhores resultados, ela o fez bloqueando uma maior quantidade de reservas imediatas em comparação com as reservas antecipadas. A estratégia 3 e 4 vem em seguida, equalizando mais essa diferença e alcançando bons resultados de utilização. Por fim, a estratégia 1, que embora tivesse alcançado bons resultados para as reservas antecipadas, não obteve uma boa taxa de utilização da rede.

CONSIDERAÇÕES FINAIS

Para que aplicações intensivas de dados possam ter um bom funcionamento é necessário que as mesmas possam ser capazes de reservar canais dedicados de largura de banda. Esses canais dedicados têm sido providos pelas redes de circuito dinâmico (DCN), que são redes menores e possuem o seu funcionamento diferente da tradicional Internet. A diferenciação consiste no fato de que os donos das aplicações nas DCN podem sinalizar, junto a sua requisição, restrições para o bom funcionamento da aplicação, ao invés de simplesmente submetê-la, sem nada a dizer, e aguardar o comportamento da mesma. Esta dissertação teve como objetivo investigar os diferentes modelos de requisição de recursos para as redes de circuito dinâmico e o seu respectivo algoritmo de alocação de largura de banda.

No âmbito das redes de circuito dinâmico, um tipo de aplicação bastante comum é a transferência de arquivos na ordem de terabytes. Esse tipo de aplicação se caracteriza por apresentar um alto grau de flexibilidade, tendo em vista que permite a transmissão dos dados em diversas ordens de transferência. Nas DCN geralmente apenas um modelo de reserva é comumente utilizado: a reserva antecipada. Nesse modelo, existe um desacoplamento entre o momento em que a requisição solicita o recurso e o momento em que de fato a requisição faz uso do mesmo. Além disso, o modelo de requisição dessas reservas é a *requisição fixa*, em que a aplicação define o tempo de início e fim da transmissão, e a taxa de largura de banda utilizado. Como já explicado, esse tipo de requisição tende a criar instante de tempo com muita ou pouca requisição, além de contribuir para a fragmentação da largura de banda. Por isso, desde o início, este trabalho pretendia fazer uso da *requisição maleável*, as quais o dono da aplicação define um intervalo dentro do qual seja aceita que a transmissão se conclua, dando desse modo a oportunidade ao gerenciador do recurso definir os instantes mais apropriados para atender a demanda. Como as aplicações que fazem uso das reservas antecipadas são normalmente aplicações elásticas, seria perfeitamente possível permitir variação na taxa de transferência. Por isso, utilizou-se o modelo de *requisição elástica*.

No decorrer do trabalho, percebeu-se que na prática é muito provável que existam requisições que não puderam ser programadas com antecedência. Por exemplo, um servidor

que faz sincronismo sempre que atinge um certo limiar de diferença. Não se sabe exatamente qual o instante em que esse limiar será alcançado. Assim, diferentemente do que ocorre nas DCN, passou-se a assumir a existência de ambos tipos de reserva: imediata e a antecipada. Além disso, considerou-se um perfil de aplicações para cada tipo de reserva. As reservas antecipadas passaram a ser representadas por aplicações elásticas, e as reservas imediatas passaram a ser representadas pelas aplicações inelásticas.

Nesse contexto de permitir o uso de banda variada por parte de aplicações elásticas, o algoritmo mais utilizado é o *Quick-Finish*. Contudo, como já foi mostrado no decorrer dessa dissertação, ao colocá-lo em contato com aplicações inelásticas do tipo streaming, por exemplo, o QF acaba bloqueando uma quantidade significativa dessas requisições. Foi sob essa perspectiva, que o algoritmo Balanceamento de Banda Residual foi concebido. Um algoritmo que procura alocar mais recursos onde há mais disponibilidade e menos onde existe pouca disponibilidade. Como consequência, o BBR conseguiu harmonizar a coexistência de ambos os tipos de aplicações, elástica e inelástica, aumentando a taxa de utilização da rede. Os resultados mostraram que as maiores diferenças entre os algoritmos BBR e QF acontecem para cargas leves e moderadas (0.5 e 0.75 Erlangs). Para cargas leves, o BBR obteve taxas de aceitação total em torno de 40% maiores que o QF. Vale ressaltar que grande parte dessa porcentagem se deve a aceitação das aplicações inelásticas que encontram dificuldades de serem aceitas pelo algoritmo QF. Com isso, o algoritmo BBR chegou a alcançar até 30% a mais de utilização dos recursos. Outro fator importante é a flexibilidade das requisições. Percebeu-se que o aumento da flexibilidade até certo ponto aumenta a aceitação das requisições, porém chega um momento que não faz mais tanta diferença aumentar a flexibilidade.

Por fim, foi proposta uma estratégia modificada da literatura para o gerenciamento de recurso denominada de Particionamento Dinâmico com Preempção. Tal estratégia é independente do algoritmo de alocação de largura de banda, o que permite usá-la tanto com o algoritmo *Quick-Finish* quanto com o Balanceamento de Banda Residual. Os resultados obtidos mostraram uma diminuição na diferença de probabilidade de aceitação entre reservas antecipadas e imediatas, além de obter uma boa utilização da rede. Supõe-se que os resultados só não foram melhores porque não se aplicou uma heurística para decidir quais reservas antecipadas, passíveis de interrupção, seriam removidas.

Tanto a proposta de alocação de largura de banda quanto a de gerenciamento de recursos podem ser implementadas nos frameworks que gerenciam as redes de circuito dinâmico, desde que sejam compatíveis com as premissas utilizadas nesta dissertação.

6.0.1 Trabalhos Futuros

Foi utilizado como topologia para os experimentos a topologia NFSnet. É importante fazer uma investigação mais profunda, no sentido de identificar qual o comportamento do algoritmo proposto com as mais diferentes topologias de rede.

Um outro ponto que pode ser explorado são os caminhos escolhidos, ao reservar recursos para as reservas imediatas. No atual trabalho, os k menores caminhos são analisados em uma certa sequência, em que o primeiro caminho que tiver recursos para aceitar a requisição é o escolhido. Na chegada de uma nova requisição, os caminhos a serem ana-

lisados começam a partir do último aceite, de modo que a última tentativa seja feita no último caminho escolhido. Em um trabalho futuro, seria interessante levar em consideração outros elementos para escolha do caminho, como os instantes de término das reservas imediatas e antecipadas já planejadas no sistema.

Por fim, outro tópico a ser explorado está relacionado a qual requisição remover em caso de falta de recursos. Na pesquisa atual, a última requisição de cada instante de tempo que coincide com o intervalo da requisição imediata é sempre a requisição a ser removida. Em futuros trabalhos, pode-se fazer uma remoção mais seletiva, por exemplo, removendo a requisição que utiliza muitos recursos, ou a requisição que possui um prazo de término mais longo na tentativa de resubmetê-la ao sistema e etc. O objetivo é interromper o menor número de reserva antecipada de modo a conseguir aceitar a reserva imediata.

REFERÊNCIAS BIBLIOGRÁFICAS

- ATT. 2013. Disponível em: <<http://www.att.com/>>. Acesso em 26/04/2013.
- AUTOBAHN. 2013. Disponível em <<http://geant2.archive.geant.net/server/show/conWebDoc.1018.html>>. Acesso em 26/04/2013.
- BALMAN, M. Advance resource provisioning in bulk data scheduling. In: *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications*. [S.l.: s.n.], 2013.
- BALMAN, M. et al. A flexible reservation algorithm for advance network provisioning. In: *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for*. [S.l.: s.n.], 2010. p. 1–11.
- BORST, S. C.; MITRA, D. Virtual partitioning for robust resource sharing: Computational techniques for heterogeneous traffic. *IEEE J.Sel. A. Commun.*, IEEE Press, Piscataway, NJ, USA, v. 16, n. 5, p. 668–678, set. 2006. ISSN 0733-8716. Disponível em: <<http://dx.doi.org/10.1109/49.700904>>.
- BURCHARD, L.-O.; HEISS, H.-U.; ROSE, C. D. Performance issues of bandwidth reservations for grid computing. In: *Computer Architecture and High Performance Computing, 2003. Proceedings. 15th Symposium on*. [S.l.: s.n.], 2003. p. 82–90.
- CANARIE. 2013. Disponível em: <<http://www.canarie.ca/en/home>>. Acesso em 26/04/2013.
- CHARBONNEAU, N. et al. Advance reservation frameworks in hybrid ip-wdm networks. *Communications Magazine, IEEE*, v. 49, n. 5, p. 132–139, 2011. ISSN 0163-6804.
- CHARBONNEAU, N.; VOKKARANE, V. M. A survey of advance reservation routing and wavelength assignment in wavelength-routed wdm networks. *Communications Surveys Tutorials, IEEE*, v. 14, n. 4, p. 1037–1064, 2012. ISSN 1553-877X.
- EI-KADI, M.; OLARIU, S.; ABDEL-WAHAB, H. Rate-based borrowing scheme for qos provisioning in multimedia wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, v. 13, n. 2, p. 156–166, Feb 2002. ISSN 1045-9219.
- ESCALONA, E. et al. Advance reservations for service-aware gmpls-based optical networks. *Computer Networks*, v. 52, n. 10, p. 1938–1950, 2008. Disponível em: <<http://dblp.uni-trier.de/db/journals/cn/cn52.html/#EscalonaSCJ08>>.
- ESNET. 2013. Disponível em: <<http://www.es.net/>>. Acesso em 26/04/2013.

ESNET. 2015. Disponível em: <<https://www.es.net/engineering-services/oscars/how-it-works/>>. Acesso em 17/02/2015.

EXTREME, J. N. 2013. Disponível em: <http://www.jgn.nict.go.jp/english/reports/presentation/documents/apii_ws-2012_01.pdf>. Acesso em 27/04/2013.

GUERIN, R.; ORDA, A. Networks with advance reservations: the routing perspective. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [S.l.: s.n.], 2000. v. 1, p. 118–127 vol.1. ISSN 0743-166X.

HE, E.; WANG, X.; LEIGH, J. A flexible advance reservation model for multi-domain wdm optical networks. In: *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*. [S.l.: s.n.], 2006. p. 1–10.

INTERNET2. 2010. Disponível em: <<http://www.internet2.edu/pubs/DCN-howto.pdf>>. Acesso em 26/04/2013.

INTERNET2. 2013. Disponível em: <<http://www.internet2.edu/network/>>. Acesso em 26/04/2013.

JANET. 2013. Disponível em: <<https://www.ja.net/products-services/janet-research/lightpath>>. Acesso em 26/04/2013.

LAI, F.; MISIC, J.; CHANSON, S. Complete sharing versus partitioning: quality of service management for wireless multimedia networks. In: *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on*. [S.l.: s.n.], 1998. p. 584–593. ISSN 1095-2055.

MALLA, A.; EL-KADI, M.; TODOROVA, P. A fair resource allocation protocol for multimedia wireless networks. In: *Parallel Processing, 2001. International Conference on*. [S.l.: s.n.], 2001. p. 437–443. ISSN 0190-3918.

NAIKSATAM, S.; FIGUEIRA, S. Elastic reservations for efficient bandwidth utilization in lambdagrids. *Future Gener. Comput. Syst.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 23, n. 1, p. 1–22, jan. 2007. ISSN 0167-739X. Disponível em: <<http://dx.doi.org/10.1016/j.future.2006.02.013>>.

NETTO, M. A.; BUBENDORFER, K.; BUYYA, R. Sla-based advance reservations with flexible and adaptive time qos parameters. In: *Proceedings of the 5th international conference on Service-Oriented Computing*. Berlin, Heidelberg: Springer-Verlag, 2007. (ICSOC '07), p. 119–131. ISBN 978-3-540-74973-8. Disponível em: <http://dx.doi.org/10.1007/978-3-540-74974-5_10>.

OSCARS. Janeiro, 2014. Disponível em: <<http://www.es.net/services/oscars/>>. Acesso em 26/04/2013.

PATEL, A. et al. Routing and scheduling for variable bandwidth advance reservation in elastic applications. In: *Optical Fiber Communication - includes post deadline papers, 2009. OFC 2009. Conference on*. [S.l.: s.n.], 2009. p. 1–3.

PATEL, A.; JUE, J. Routing and scheduling for variable bandwidth advance reservation. *Optical Communications and Networking, IEEE/OSA Journal of*, v. 3, n. 12, p. 912–923, 2011. ISSN 1943-0620.

RAJAH, K.; RANKA, S.; XIA, Y. Advance reservations and scheduling for bulk transfers in research networks. *Parallel and Distributed Systems, IEEE Transactions on*, v. 20, n. 11, p. 1682–1697, 2009. ISSN 1045-9219.

RNP. 2013. Disponível em: <<http://www.rnp.br>>. Acesso em 28/04/2013.

ROUSSEAU, D.; TRIAY, J.; VOKKARANE, V. Improving service differentiation of immediate and advance reservation in resource-partitioned optical wdm networks. In: *Computing, Networking and Communications (ICNC), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 173–179.

SCHMIDT, S.; KUNEGIS, J. Scalable bandwidth optimization in advance reservation networks. In: *Networks, 2007. ICON 2007. 15th IEEE International Conference on*. [S.l.: s.n.], 2007. p. 95–100. ISSN 1556-6463.

SHARMA, S.; KATRAMATOS, D.; YU, D. End-to-end network qos via scheduling of flexible resource reservation requests. In: *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. New York, NY, USA: ACM, 2011. (SC '11), p. 68:1–68:10. ISBN 978-1-4503-0771-0. Disponível em: <<http://doi.acm.org/10.1145/2063384.2063475>>.

SHEN, L. et al. A two-phase approach for dynamic lightpath scheduling in wdm optical networks. In: *Communications, 2007. ICC '07. IEEE International Conference on*. [S.l.: s.n.], 2007. p. 2412–2417.

SURFNET. 2013. Disponível em: <<http://www.surfnet.nl/en/Pages/default.aspx>>. Acesso em 26/04/2013.

UCLP. 2013. Disponível em: <<http://www.i2cat.net/en>>. Acesso em 18/11/2013.

VADREVU, C. et al. Hybrid circuit/packet networks with dynamic capacity partitioning. In: *Kaleidoscope: Beyond the Internet? - Innovations for Future Networks and Services, 2010 ITU-T*. [S.l.: s.n.], 2010. p. 1–7.

VEERARAGHA, M.; JUKAN, A. 2010. Disponível em: <<http://www.ece.virginia.edu/mv/research/DOE09/documents/deliverables/feb2010/hybrid-arch-final.pdf>>. Acesso em 26/04/2013.

VERIZON. 2013. Disponível em: <<http://www22.verizon.com/>>. Acesso em 26/04/2013.

WOLF, L. C.; STEINMETZ, R. Concepts for resource reservation in advance. *Multimedia Tools Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 4, n. 3, p. 255–278, maio 1997. ISSN 1380-7501. Disponível em: <<http://dx.doi.org/10.1023/A:1009684906050>>.

XIE, C.; GHANI, N. A prioritized routing algorithm for multi-domain optical networks supporting advance reservation. In: *High Capacity Optical Networks and Enabling Technologies, 2008. HONET 2008. International Symposium on*. [S.l.: s.n.], 2008. p. 211–215.

YEN, J. Y. Finding the k shortest loopless paths in a network. *Management Science*, v. 17, n. 11, p. 712–716, 1971. Disponível em: <<http://dx.doi.org/10.1287/mnsc.17.11.712>>.

ZHU, X.; VEERARAGHAVAN, M. Analysis and design of book-ahead bandwidth-sharing mechanisms. *Communications, IEEE Transactions on*, v. 56, n. 12, p. 2156–2165, December 2008. ISSN 0090-6778.