

Políticas de Escalonamento para Transferências de Dados em Massa Inter Centro de Dados Utilizando Roteamento e Alocação de Espectro

Léia Sousa de Sousa, André Costa Drummond

¹Departamento de Ciência da Computação (CIC) – Universidade de Brasília (UnB)
Caixa Postal 4.466 – 70910-900 – Brasília – DF – Brasil

{leia.unb@gmail.com, andred@unb.br}

Abstract. *Bulk Data Transfers (BDT) and their generalization, Multiple BDT (MBDT) are critical services for the operation of businesses in the geo-distributed inter- data center networks. In these networks, BDT or MBDT applications perform synchronizations, resynchronizations, data replication, virtual machine migration and big data aggregation, among other tasks, requiring large amounts of bandwidth and efficient traffic engineering. Elastic Optical Networks (EON) allow the provisioning and allocation of resources on demand through Routing and Spectrum Allocation (RSA). This work proposes and compares dynamic application-aware RSA solutions that employ Smallest Deadline First (SDF), Larger Data Amount (LDA), Smallest Data Amount (SDA) and Smallest Remaining Time (SRT).*

Resumo. *Transferências de Dados em Massa (BDT) e sua generalização, Múltiplas BDTs (MBDT), são serviços críticos para o funcionamento dos negócios nas redes inter centro de dados geo-distribuídas. Nessas redes, aplicações do tipo BDT ou MBDT realizam sincronizações, ressincronizações, replicação de dados, migração de máquinas virtuais e agregação big data, entre outras tarefas, exigindo grandes quantidades de largura de banda e soluções eficientes de engenharia de tráfego. As Redes Ópticas Elásticas (EON) permitem o provisionamento e alocação de recursos sob demanda por meio do Roteamento e Alocação de Espectro (RSA). Este trabalho propõe e compara soluções dinâmicas de RSA cientes da aplicação, que empregam as técnicas de escalonamento Smallest Deadline First (SDF), Larger Data Amount (LDA), Smallest Data Amount (SDA) e Smallest Remaining Time (SRT).*

1. Introdução

Transferências de Dados em Massa (BDT) e sua generalização, Múltiplas BDT (MBDTs), constituem demandas básicas para o funcionamento da maioria dos serviços oferecidos pelos provedores de computação em nuvem. Nas redes inter centro de dados (ICD), principalmente das conexões inter continentais, as requisições para realizar sincronizações, ressincronizações, replicação, migração de máquinas virtuais e agregação *big data* são escalonadas por nível de urgência para aproveitar de maneira elástica os recursos de largura de banda disponíveis entre os centros de dados (CD) [Li et al. 2018, Lu and Zhu 2015, Lu et al. 2017].

Enquanto BDT representa uma requisição de serviço individual (Figura 1(a)), MBDT representa um lote (*batch*) de requisições do tipo BDT (Figura 1(b)). Tipicamente, BDT são usadas para migrar de máquinas virtuais e *backups* ICD para replicação de grandes conjuntos de dados, entre outras finalidades, ao passo que MBDT pode ser adotada para o transporte de *chunks* de dados, isto é, subpartes de um BDT que precisou ser dividido para aproveitar a banda da rede que era insuficiente para enviá-lo por completo antes da divisão [Wu et al. 2017]. MBDT também são intensamente utilizados para efetuar resincronizações ICD, o que consiste na atualização de uma dada réplica de dados pertencente a um conjunto de réplicas distribuídas geograficamente para prover consistência de dados, disponibilidade e diminuição da latência no acesso a esses dados [Castro and Liskov 2002, Lu et al. 2017, Li et al. 2018].

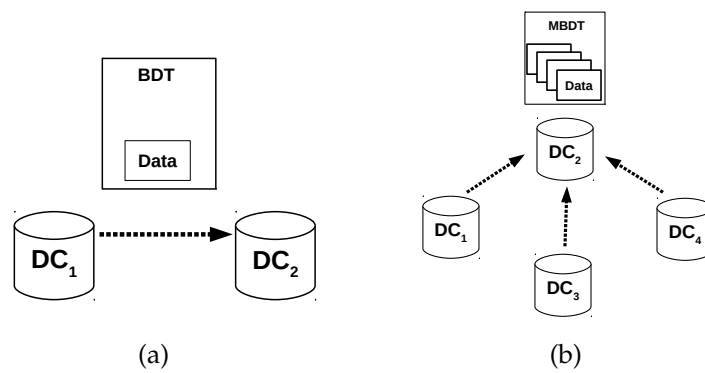


Figura 1. Representação das operações de (a) BDT e (b) MBDT.

Os principais desafios associados a esses tipos de aplicações são: (i) o tamanho do *bulk*, que pode variar de dezenas de *gigabytes* a *terabytes* de dados; (ii) a garantia do *deadline* da transferência, isto é, providenciar para que o volume chegue inteiramente ao destino antes ou até um dado limite de tempo previamente estipulado, o que é muito mais complexo de ocorrer quando se trata de MBDT devido a quantidade de chamadas desse *batch*; (iii) configurar conexões de maneira dinâmica. Atualmente, uma vez estabelecidas, as conexões permanecem ativas por meses ou anos, pois o projeto da camada óptica foi concebido para constante conexões de taxa de *bits* que não exijam configuração de conexão rápida; e (iv) garantir atendimento relativamente justo para diferentes tipos de serviços (classes de serviços) sendo todos famintos por largura de banda [Zhang et al. 2015, Luo et al. 2018, Wu et al. 2017, Srinivasan et al. 2018].

A tecnologia das Redes Ópticas Elásticas (EON) tem sido destacada como a nova geração de redes ópticas de transporte. A arquitetura é composta por *transponders* de largura de banda variável (BVTs) e comutadores ópticos de largura de banda variável (BV-OXC), que permitem eficientemente alocar uma rota adequada e largura de banda na forma de fatias (*slots*) do espectro óptico [Lu and Zhu 2015]. Os BVTs convertem o sinal de dados do domínio elétrico para o domínio óptico e ajustam a modulação para definir a taxa de transferência. Os BV-OXCs estabelecem a rota fim-a-fim para o encaminhamento do tráfego. A alocação de um caminho viável com largura de banda suficiente é feita resolvendo-se o problema

de roteamento e alocação de espectro (RSA). Para resolver o problema RSA é preciso atender a quatro restrições que aumentam ainda mais a complexidade: (i) a restrição de continuidade garante que os mesmos *slots* de frequência sejam alocadas em cada enlace do caminho; (ii) a restrição de contiguidade assegura que no caso da alocação de mais de um *slot*, estes devem ser adjacentes; (iii) a restrição de banda de guarda suficiente, exige alguns *slots* disponíveis entre quaisquer duas conexões estabelecidas para garantir que o sinal transmitido seja corretamente detectado no destino; (iv) a não sobreposição de espectro garante que duas conexões não ocupem simultaneamente a mesma porção do espectro óptico [Lu and Zhu 2015, Wang et al. 2015].

Entretanto, o oferecimento de nova arquitetura de rede e aumento da capacidade de transmissão não são suficientes para assegurar que seja possível lidar com as futuras demandas de tráfego, principalmente quando algumas classes de tráfego requerem critérios rígidos de atendimento em detrimento de outras [Luo et al. 2018, Wu et al. 2017, Aibin and Walkowiak 2017, Deng and Wang 2018]. É necessário investigar soluções eficientes de engenharia de tráfego para melhorar o atendimento dos estritos requisitos de Qualidade de Serviço (QoS). Nesse contexto se aplicam as políticas de escalonamento de requisições, por meio das quais os serviços que chegam na rede obtêm mais oportunidades de encontrar recursos disponíveis, mesmo quando se tratar de tráfego dinâmico. Escalonamentos são implementados através da classificação de requisições por um dado atributo da classe de tráfego como, por exemplo, volume de dados transferidos ou prazo para transferência [Wu et al. 2017, Lu and Zhu 2015, Aibin and Walkowiak 2017, Deng and Wang 2018]. Outros critérios adotados principalmente em ambiente de tráfego dinâmico são reserva imediata (IR) e reserva antecipada (AR) de recursos [Wang et al. 2015, Lu and Zhu 2015]. Outro ponto que tem sido explorado é a adoção de roteamento ciente da aplicação (AA)[Deng and Wang 2018], no qual informações da camada de aplicação (classificação do tráfego, métricas de QoS, entre outras) são utilizadas como entrada para a tomada de decisões na camada de rede.

Neste trabalho propõem-se quatro soluções de escalonamento para atendimento das requisições BDTs e MBDTs que são classificadas pelas seguintes políticas: *Smallest Deadline First* (SDF), *Larger Data Amount* (LDA), *Smallest Data Amount* (SDA) e *Smallest Remaining Time* (SRT). Quando uma requisição chega, ocorre a primeira tentativa de atendimento. Caso não seja possível atendê-la nesse tempo, devido à indisponibilidade de recursos, a requisição é encaminhada para a janela de escalonamento onde uma das políticas é aplicada para classificação. Em todos os casos a seleção de recursos é feita por meio de um algoritmo RSA ciente da aplicação (AA), isto é, que reconhece a relação entre as requisições individuais dentro de um lote (*batch*) e utiliza uma estratégia específica para atribuição de espectro para diferentes classes.

Os resultados obtidos nos experimentos com tais políticas, mostram que a escolha de alguma estratégia de escalonamento é suficiente para postergar o tempo no qual uma requisição seria bloqueada. Além disso, os melhores resultados são obtidos com a política SRT, que consegue atender a mais de 80% das

requisições BDTs e MBDTs submetidas na rede. Adicionalmente, quando a rede está sobrecarregada (mais de 18 chegadas de requisições por unidade de tempo), as políticas que priorizam as requisições de menor granularidade (tanto de quantidade de dados quanto de *deadline*) conseguem manter altas taxas de aceitação (além dos 60%) para todas as demandas.

O restante deste trabalho está organizado da seguinte maneira. Na Seção 2 é apresentado um levantamento da literatura com foco em políticas de escalonamento e soluções de roteamento e alocação de recursos para diferentes tipos de tráfego. Na Seção 3 são apresentadas as soluções algorítmicas propostas. Na Seção 4 são mostrados os resultados alcançados com as soluções adotadas. Finalmente, na Seção 5 são apresentadas as considerações finais.

2. Trabalhos Relacionados

As soluções de escalonamento e seleção de recursos que existem na literatura são exploradas de diversas maneiras, e o principal objetivo que motiva esse tópico é a maximização da eficiência no uso de recursos, muito vezes escassos diante das crescentes demandas. As propostas destacadas a seguir são direcionadas para EON ([Wang et al. 2015, Lu and Zhu 2015, Lu and Zhu 2015, Lu et al. 2017]), para redes definidas por *software* (SDN) ([Wu et al. 2017, Li et al. 2018]) ou sua combinação ([Aibin and Walkowiak 2017]).

Em [Wang et al. 2015] é apresentada uma proposta para AR por meio de um algoritmo RMSA em duas etapas que, na primeira etapa verifica recursos disponíveis na dimensão do espectro e, na segunda etapa, verifica disponibilidade na dimensão do tempo para que ocorra o agendamento. O objetivo é transferir a maior quantidade de dados possível. Com esse mesmo objetivo, [Lu and Zhu 2015] apresenta uma solução otimizada RSA que permite AR e IR. As requisições BDT são escalonadas para atendimento com os recursos do espectro que sobram após o atendimento de requisições de maior prioridade. As requisições são ordenadas em ordem decrescente de quantidade de banda requerida em termos de número de *slots* do espectro. Já [Lu et al. 2017] realiza o escalonamento de BDTs e MBDT (requisições intensivas em dados) inter centro de dados em EON através de um *framework* com Funções de Virtualização da Rede (vNF). A maior prioridade é dada para as chamadas que podem ser servidas com menor tempo para, dessa maneira, minimizar o tempo de conclusão do serviço.

Considerando o paradigma SDN, [Wu et al. 2017] foca no escalonamento de BDT usando a política *store-and-forward* (SnF). Nessa política, as transferências ocorrem nos horários de maior disponibilidade de recursos na rede (preferencialmente, durante a madrugada), e o fuso-horário dos *sites* são sempre levados em consideração na tarefa de admissão do serviço. Adicionalmente, o volume de dados de um BDT é subdividido e enviado por múltiplos caminhos para aumentar as chances de conseguir largura de banda, e conseqüentemente, maximizar a taxa de aceitação de requisições. Outra proposta que aborda SDN para escalar MBDT é encontrada em [Li et al. 2018]. As MBDT são requisições a partir de múltiplas fontes, mas que não são tratadas em lote (*batch*), como no caso da proposta deste trabalho. Um sistema de gerenciamento de tráfego é apresentado para

fazer o agendamento do tempo de atendimento para as requisições. A política de escalonamento procura os caminhos com as maiores disponibilidades de acordo com as demandas no eixo do tempo. Com isso, o sistema consegue maximizar a utilização da rede.

Uma solução combinada de EON reunido com SDN é explorada em [Aibin and Walkowiak 2017]. Nessa solução, que escalona recursos ao invés de requisições, um controlador SDN mantém informações sobre recursos internos dos centro de dados (como quantidade de núcleos de processamento, memória e armazenamento de uma máquina virtual) e preços dos serviços oferecidos (como máquinas virtuais e serviços de armazenamento em massa, *Store-as-a-Cloud-SaaS*). Os recursos determinados pelo RMSA são combinados com os serviços solicitados aos CDs. As políticas de escalonamento adotadas são: escolher o CD mais próximo do nó solicitante, o mais barato em termos de custo pelo serviço e o menos solicitado (com menos recursos alocados).

3. Soluções Propostas

Baseado na literatura recente de escalonamento das requisições intensivas em dados (BDT e MBDT) [Aibin and Walkowiak 2017, Wang et al. 2015, Lu and Zhu 2015, Lu and Zhu 2015], este trabalho propõe soluções de RSA dinâmicas em EON, a qual pode ser modelada como um grafo direcionado $G(V, E)$ onde V é o conjunto de nós ópticos equipados com BV-OXC e BVT, e E é o conjunto de enlaces de fibra óptica. No conjunto V , alguns nós ópticos estão diretamente conectados a CDs e serão selecionados como origem e destino das requisições de transferência de dados.

A largura de banda de cada enlace $e \in E$ é dividida na dimensão do espectro em N slots (FS) de mesma granularidade, tipicamente 12.5GHz. O estabelecimento de um caminho óptico para atender uma requisição ocorre com a seleção dos mesmos slots em cada enlace que compõe o caminho. Para consultar a disponibilidade e realizar alocação, esses slots são numerados de 1 até $|N|$ e então são aplicadas políticas como, por exemplo, *first-fit* (varre os slots na ordem crescente) e *last-fit* (varre os slots na ordem decrescente).

Na abordagem de alocação adaptativa de espectro [Chatterjee et al. 2017] difentes políticas podem ser utilizadas de acordo com critérios das requisições ou condições da rede. A utilização das políticas *first-fit* e *last-fit* reduzem a dispersão das conexões. Com a reserva dos slots disponíveis, e segundo um dado nível de modulação O-OFDM, define-se a eficiência espectral na qual a transmissão ocorrerá, ou seja, a quantidade de bits por símbolos para cada slot ($b/s/Hz$) [Lu and Zhu 2015].

Considera-se o conjunto \mathfrak{R} formado por demandas de dois tipos, BDT e MBDT, que chegam na rede solicitando atendimento. Essas demandas podem ser servidas imediatamente ou são escalonadas para nova tentativa de atendimento no futuro. Requisições BDT são modeladas como $r = (s_r, d_r, C_r, dl_r)$, onde s_r e d_r são origem e destino respectivamente, entre os quais uma quantidade de dados C_r é transferida dentro de um prazo dl_r com taxa mínima, $\beta_{min}^r = \frac{C_r}{dl_r}$. Já as requisições em *batch*, MBDT, são representadas como $R = \{r_1, r_2, \dots, r_n\}$ onde

$dl_{r_1} = dl_{r_2} = \dots = dl_{r_n}$. A taxa a ser atribuída a R equivale a taxa máxima, β_{max}^R , que é a máxima capacidade de banda disponível no canal no momento do atendimento. Como o roteamento é ciente da aplicação MBDT, esse tipo de requisição é atendida quando o mínimo de três BDT de um *batch* consegue recursos. A quantidade de réplicas de dados praticada no mercado hoje é de três bases distribuídas [Aibin and Walkowiak 2017, Luo et al. 2018, Wu et al. 2017]. Isso significa que, em um *batch* com mais de três requisições, o excedente desse grupo é descartado, uma vez que, no caso de uma ressincronização de uma única partição de dados, é preciso assegurar o limite de tolerância a falhas Bizantinas [Castro and Liskov 2002]. Atender a mais do que três sub-requisições resulta no mesmo efeito de atualizar a partição, mas leva ao desperdício de recursos.

O escalonamento das demandas \mathfrak{R} oferece mais oportunidades para que as requisições competindo por recursos sejam atendidas, até o prazo especificado. Os provedores de rede exploram políticas de escalonamento para garantir o atendimento do QoS e diminuir a taxa de bloqueio de requisições. Neste trabalho, as estratégias de escalonamento de requisições serão as seguintes:

- Menor prazo primeiro (SDF): ordena as requisições pela ordem crescente de dl , pois assim, o provedor trata com prioridade as demandas que ocuparão a rede por menos tempo no pior caso, uma vez que esse atendimento pode ocorrer no tempo t , tal que $0 \leq t \leq dl$ [Lu et al. 2017].
- Maior quantidade de dados a ser transferida (LDA): ordena as requisições em ordem decrescente de quantidade de dados C que deverão ser transferidos. Os provedores podem adotar essa técnica de priorização principalmente nos horários de mais baixa utilização dos recursos, pois dessa maneira há maiores garantias de que as demandas com maior probabilidade de comprometer o desempenho da rede, sejam inteiramente transferidas sem que seja necessário dividi-las em demandas menores [Aibin and Walkowiak 2017];
- Menor quantidade de dados a ser transferida entre a origem e destino (SDA): essa ordenação, reversa da política LDA, pode ser útil para aproveitar fragmentos do espectro óptico com banda suficiente, promovendo o uso mais eficiente dos recursos [Lu and Zhu 2015];
- Menor tempo restante para atendimento (SRT): para as requisições na fila é calculada a diferença entre o prazo total da transferência e o tempo atual. Essa diferença denota o tempo restante para que o prazo de atendimento de uma demanda se esgote. A fila é então ordenada pelo menor tempo restante para favorecer as requisições que estão próximas de atingir esse limite. Assim, desperdiça-se menos poder de processamento no roteamento, que ocorreria para tentar atender uma requisição com probabilidade iminente de bloqueio [Srinivasan et al. 2018].

Para exemplificar, considere as demandas $r_a = \{1, 3, 300GB, 80u.t.\}$, $R_b = \{\{1, 2, 3, 5\}, 4, 100GB, 10u.t.\}$ e $r_c = \{4, 2, 500GB, 20u.t.\}$. Um algoritmo de alocação dinâmica que não realiza escalonamento, tentará encontrar recursos para atendê-las e caso não haja banda disponível, essas requisições são logo bloqueadas, ainda que esteja dentro do prazo de atendimento. Entretanto, ao adotar qualquer política de escalonamento, a probabilidade de bloqueio das requisições pode

ser reduzida porque essas requisições podem ser submetidas mais de uma vez. Quando a política SDF é escolhida, R_b é atendida inicialmente, e mesmo que seja uma demanda de maior volume, as requisições subsequentes serão r_c e r_a , que poderão aguardar na janela de escalonamento até que os recursos estejam livres. Com a política LDA, a ordem de atendimento é r_c (500GB), R_b (4 demandas de 100GB, ou seja, 400GB) e, finalmente, r_a (300GB). Por outro lado, com a política SDA, o atendimento é inicialmente dedicado para a requisição r_a , e somente depois para R_b e r_c . Finalmente, com a política de escalonamento SRT, é preciso considerar o período de tempo atual no relógio da simulação, para assim, periodicamente, computar o tempo restante de cada requisição.

3.1. Algoritmos de Escalonamento de Requisições em EON

O Escalonamento Dinâmico de Demandas (Algoritmo 1) é empregado para tratar as demandas BDTs e MBDT, para as quais as tarefas de roteamento e alocação de espectro são efetuadas pelo Algoritmo RSA($G, k, demanda$) (Algoritmo 2). Para cada simulação com um conjunto de execuções, o Algoritmo 1 implementa, na linha 6, uma das seguintes políticas de priorização das demandas: SDF, LDA, SDA ou SRT. Já o Algoritmo 2 procura atender uma demanda BDT com taxa mínima β_{min} na ordem ascendente dos *slots* de espectro (política *first-fit*) na linha 5, enquanto que as demandas MBDTs são servidas, na linha 10, na ordem descendente dos *slots* de frequência (política *last-fit*).

O Algoritmo 1 recebe como entrada o conjunto de demandas \mathfrak{R} , a topologia da rede G , a quantidade k de caminhos candidatos para estabelecimento de uma conexão e um valor inteiro referido como *FatorBizantino* que designa o número de réplicas utilizadas para atualizar uma dada partição em um CD de destino. A janela de escalonamento W (fila de prioridade) é inicializada na linha 2. Cada demanda é encaminhada para a janela de escalonamento W (linha 4), e na sequência, a operação *RemoveAtrasados*(W) verifica e remove as demandas que estão com iminente probabilidade de bloqueio devido ao esgotamento do *deadline* (linha 5). Na linha 6, e de acordo com a política de escalonamento adotada, as requisições são devidamente ordenadas em ordem decrescente de prioridade. Assim, se a demanda prioritária naquele momento for do tipo MBDT, computam-se as combinações com três requisições (*FatorBizantino* = 3) para selecionar as requisições do *batch* que poderão ser atendidas (linha 9). Esse procedimento é realizado também pelo algoritmo AARSAE([Sousa and Drummond 2017]) com o qual as soluções apresentadas neste trabalho são comparadas. O atendimento do *batch* ocorre com uma chamada ao Algoritmo 2, na linha 11. Já as requisições BDTs podem ser atendidas pelo Algoritmo 2, chamado na linha 16.

O Algoritmo 2 visa atender as demandas de MBDT com taxa máxima β_{max} para reduzir o tempo que esse tipo de requisição permanece ocupando a rede. Como a ressincronização representa o encaminhamento de dados de múltiplas origens para um dado destino, é importante que a banda nos enlaces desse trajeto sejam liberadas para reduzir o congestionamento em ocasiões de tráfego pesado.

Algoritmo 1 Escalonamento Dinâmico de Demandas

```
1: Entradas:  $\mathfrak{X}, G, k, \text{FatorBizantino}$ 
2:  $W \leftarrow \emptyset$ 
3: para cada demanda  $\in \mathfrak{X}$  faça
4:   Encaminha a demanda para  $W$ 
5:   RemoveAtrasados( $W$ )
6:   AplicaPolíticaDeEscalonamento( $W$ )
7:   para cada demanda  $\in W$  faça
8:     se demanda é  $R$  então                                     ▶ Demanda do tipo MBDT
9:       ObtemCombinações( $R, \text{FatorBizantino}$ )
10:      para cada combinação de  $R$  faça
11:        se  $\text{RSA}(G, k, R)$  é verdadeiro então
12:          retorne
13:        fim se
14:      fim para
15:    senão                                                         ▶ Demanda do tipo BDT
16:      se  $\text{RSA}(G, k, r)$  é verdadeiro então
17:        retorne
18:      fim se
19:    fim se
20:  fim para
21: fim para
```

Algoritmo 2 $\text{RSA}(G, k, \text{demanda})$

```
1:  $P \leftarrow \text{KSP}(k, \text{demanda})$ 
2: Ordena  $P$  em ordem decrescente de disponibilidade de banda
3: para cada  $p \in P$  faça
4:   se a demanda é  $r$  então
5:     se  $\beta_{\min}$  pode ser servida em  $p$  com a política first-fit então
6:       Aceita  $r$ 
7:       Retorne Verdadeiro
8:     fim se
9:   senão
10:    se  $\beta_{\max}$  pode ser servida em  $p$  com a política last-fit então
11:      Aceita  $R$ 
12:      Retorne Verdadeiro
13:    fim se
14:  fim se
15: fim para
```

A grande vantagem de utilizar a janela de escalonamento W é a possibilidade de realizar mais de um tentativa de atendimento das demandas antes que seu *deadline* prescreva, uma vez que os bloqueios de requisição só ocorrem por este motivo. Dessa maneira, é possível que a taxa de bloqueio seja reduzida em comparação com algoritmos que não realizam escalonamento.

3.2. Análise da Complexidade

O Algoritmo de Escalonamento Dinâmico de Demandas (Algoritmo 1) lida com requisições BDTs e MBDTs. A inicialização da janela de escalonamento, na linha 2 toma um tempo constante. O laço da linha 3 até a linha 21 é executado para cada demanda de \mathfrak{R} que chega na rede, e ao chegar é encaminhada para W (linha 4), o que ocorre em tempo constante. A operação *RemoveAtrasados*(W) (linha 5) requer que toda a janela seja percorrida para que seja possível remover uma requisição da fila de prioridade, tomando, para isso, $O(|\mathfrak{R}|)$.

A operação *AplicaPolíticaDeEscalonamento*(W) define a política (SDF, LDA, SDA ou SRT), que é adotada para organizar a fila de prioridades. Em todo caso, a ordenação é feita olhando-se para os atributos específicos da requisição, como C ou dl , para cada uma das requisições. A fila de prioridades pode ser modelada como um *heap* binário. As ordenações SDF, SDA e SRT são implementadas usando *heap-Min*, enquanto que LDA é feita com *heap-Max*. Ambas são executadas em $O(|\mathfrak{R}|\log|\mathfrak{R}|)$ para todas as requisições [Cormen et al. 2009].

O laço das linhas 7 até 20 é executado para acessar cada demanda em W , que pode ser do tipo MBDT ou BDT, referenciadas como R (linha 8) e r (linha 16), respectivamente. A operação *ObtemCombinações*($R, FatorBizantino$) encontra combinações de chamadas pertencentes a uma demanda MBDT em tempo $O(\frac{|R|!}{(FatorBizantino! - (|R| - FatorBizantino)!})$), que é exponencial. Entretanto, os valores de $|R|$ e *FatorBizantino* são suficientemente pequenos.

O Algoritmo *RSA*($G, k, demanda$) (Algoritmo 2) é invocado nas linhas 11 e 16 do Algoritmo 1, para atender as demandas BDT e MBDT. No RSA é executado o algoritmo *KSP*($k, demanda$) [Yen 1971] (linha 1) para computar os k menores caminhos sem *loops* (para o conjunto de caminhos P), o que é feito em $O(K|V|^3)$. Os caminhos obtidos na busca com o KSP são organizados em uma lista ligada e, então, ordenados pela ordem decrescente de disponibilidade de banda (linha 2 do Algoritmo 2). Para que essa ordenação ocorra, a lista de caminhos de tamanho k é percorrida e cada caminho é comparado com os demais $k - 1$, o que resulta em complexidade de $O(K^2)$.

Para cada um dos k caminhos é feita a verificação de disponibilidade da banda solicitada e, em caso de sucesso, ocorre a alocação do espectro, conforme o laço das linhas 3 até 15 do Algoritmo 2. Para tanto, duas políticas de alocação são adotadas, *first-fit* para atender BDT (linha 5) e *last-fit* para MBDT (linha 10). A operação de detecção e intersecção de espectro nas duas políticas possui complexidade de tempo de $O(|K|^2)$, uma vez que a principal diferença entre os dois métodos é o sentido da varredura no espectro de frequência. Dessa maneira, temos que a complexidade de tempo do Algoritmo 2 é de $O(K|V|^3) + K^2 + 2K^3$. Como resultado, cada uma das políticas de escalonamento possui uma complexidade de tempo de execução de $O(|\mathfrak{R}|((K|V|^3) + K^3))$.

4. Avaliação de Desempenho

Na avaliação dos algoritmos propostos, que foram implementados no simulador ONS [Costa et al. 2016], eventos dinâmicos de chegadas e partidas de requisições

foram simuladas na rede NSFNET (Figura 2) para comparar o desempenho com o Algoritmo AARSAE [Sousa and Drummond 2017]. Na figura abaixo os nós são BV-OXCs, as arestas estão numeradas com as respectivas distâncias e tem-se alguns dos nós destacados por terem conexão direta com CD que efetuam ou recebem uma transferência de dados.

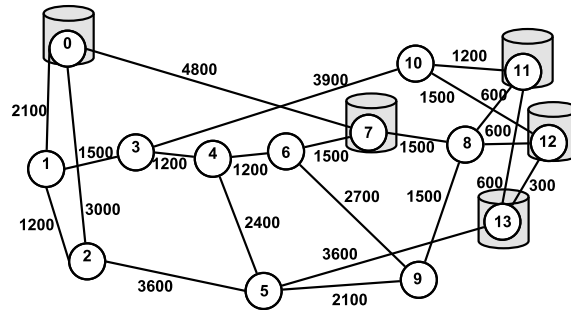


Figura 2. Topologia da rede NSFNET

Considerou-se o uso ilimitado de BVTs, sendo cada um com capacidade de transmissão de 20 slots. Cada slot tem largura de banda de 12.5GHz e cada enlace possui 640 slots de frequência com espectro total de 8THz. Para banda de guarda assumiu-se dois slots. Os algoritmos foram testados usando a modulação 8QAM [Aibin and Walkowiak 2017].

Cada simulação foi realizada 5 vezes utilizando o método de replicações independentes. Para os resultados apresentados foram calculados intervalos de confiança com 95% de confiabilidade. Foram geradas 100.000 chamadas por simulação de acordo com a distribuição de Poisson, considerando-se os dois tipos de aplicação em questão, com origens e destinos distribuídos uniformemente dentro do subconjunto de localizações dos CDs. O número de chegadas variou entre 2 e 30 por unidade de tempo [Zhang et al. 2015] com incrementos de 4 unidades. As BDTs foram configuradas para transferir [100, 300, 500, 800]GB dentro de um prazo de [20, 60, 80, 40] unidades de tempo, respectivamente. A mesma configuração das BDTs foi adotada para as MBDTs, definidas como um total de 4 requisições BDTs para um mesmo destino. A atribuição do espectro é feita utilizando as políticas *first-fit* e *last-fit* para BDTs e MBDTs, respectivamente. O uso do algoritmo KSP pelos algoritmos das políticas de escalonamento considerou $k = 4$. Na avaliação de desempenho a seguir, analisam-se as taxas de sucesso do serviço MBDT (ressincronizações) e taxa de bloqueio de BDTs.

4.1. Apresentação dos Resultados

As políticas de escalonamento de requisições SDF, LDA, SDA e SRT foram implementadas e comparadas com o algoritmo AARSAE([Sousa and Drummond 2017]). Enquanto essas políticas são usadas para escalonar os dois tipos de demandas de tráfego, BDT e MBDT, o algoritmo AARSAE não faz escalonamento e funciona da seguinte maneira: quando a rota a ser estabelecida é menor do que a metade do diâmetro da rede, o tráfego por essa rota ocorrerá com taxa máxima.

O gráfico da Figura 3 apresenta os resultados da avaliação da qualidade de atendimento do serviço por número de requisições em cada unidade de tempo. Essa avaliação é referida como taxa de sucesso de ressincronizações, que é a razão entre o número de demandas MBDTs atendidas e o número total de MBDTs. A política com melhor desempenho nesse cenário é a SRT, que ordena as requisições escalonadas pelo menor tempo restante, mantendo mais de 80% de aceitação de requisições quando ocorrem 30 chegadas de requisições por unidade de tempo. Esse fenômeno ocorre porque as chamadas sempre serão submetidas a uma tentativa de alocação antes de serem bloqueadas de fato, devido a violação do *deadline*. No caso das demais políticas, mesmo que o procedimento de ordenação das requisições mude, existe uma função *RemoveAtrasados(W)* (linha 5 do Algoritmo 1) que ao invés de atendê-las emergencialmente, bloqueia essas requisições para diminuir o número de demandas classificadas.

Em seguida, os próximos melhores resultados são conseguidos com os algoritmos SDA e SDF em termos de atendimento das MBDTs, mostrando que atender primeiro aquelas demandas de menor granularidade, tanto de volume de dados quanto de prazo, melhora a fluidez dos recursos. Essas demandas de menor granularidade são atendidas com a taxa máxima de transferência, em caminhos selecionados pela maior disponibilidade de recursos. Com isso, as transferências ocupam por menos tempo os canais estabelecidos nos *slots* em ordem descendente (política *last-fit*). De maneira conjunta, essas decisões de engenharia de tráfego aumentam a garantia de que exista recursos para atender as requisições em lote.

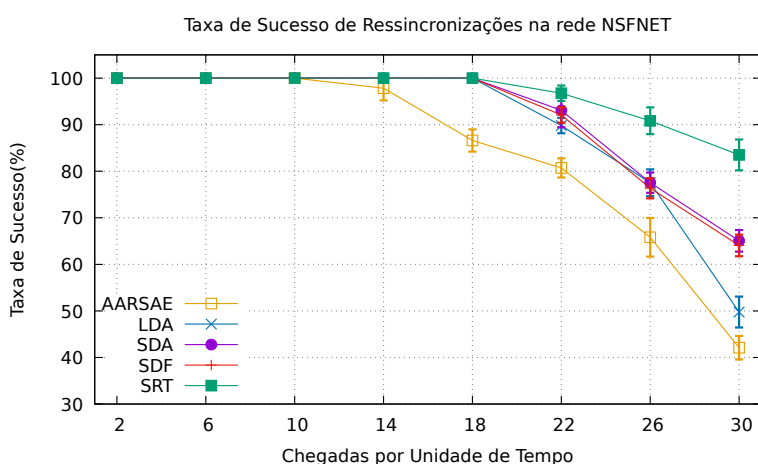


Figura 3. Taxa de sucesso de ressincronizações com as demandas MBDTs na topologia NSFNET

Já a política LDA (Figura 3), na qual as MBDTs com os maiores volumes de dados são priorizadas, alcança um percentual menor de aceitação em comparação com as duas políticas anteriores, o que é especialmente percebido no cenário onde ocorrem mais de 18 chegadas por unidade de tempo (Figura 3). Isso ocorre porque as maiores demandas tendem a congestionar o tráfego na rede mais rapidamente, um problema que é agravado pela localização relativamente próxima de um grande número de CDs, conforme mostra a Figura 2. Com isso, as chamadas esperando atendimento na janela *W* sofrem um risco maior de estapolar os

seus *deadlines* devido as tentativas mal-sucedidas de atendimento, enquanto os recursos da rede estão ocupados pelas demandas de maior granularidade. Apesar dessa dificuldade, todas as estratégias de escalonamento atingem bons resultados em comparação com o algoritmo AARSAE que não realiza escalonamento.

Já o gráfico da Figura 4, apresenta os resultados da taxa de bloqueio (BR) de requisições BDTs por número de chegadas de requisições por cada unidade de tempo. Os algoritmos AARSAE e SDF tiveram taxas de bloqueio muito próximas, com o SDF se sobressaindo em pouco menos de 0.5%. O SDF bloqueia requisições que estão no limite de tempo e prioriza as requisições com os menores prazos. O AARSAE se beneficia usando a prática de definição da taxa de transferência baseada no diâmetro da rede devido à localização dos CDs (Figura 2), concentrados em determinado ponto da topologia. Entre 18 e 22 chegadas por unidade de tempo, o SDF tem ligeira queda no desempenho em comparação com o AARSAE, mas logo essa diferença diminui.

As políticas LDA e SDA, também mostradas no gráfico da Figura 4, são as que apresentam os resultados mais similares para o bloqueio de BDTs, uma vez que ambas se baseiam no volume de dados para priorizar as demandas. Essa estratégia é tomada em um ambiente com disputa de recursos por parte de demandas BDTs e MBDTs. Como o volume de dados de uma MBDT tem maior probabilidade de superar o volume de dados de uma BDT, muitas chamadas podem ter que aguardar na janela de escalonamento por muitas tentativas consecutivas até que o seus *deadlines* estoure.

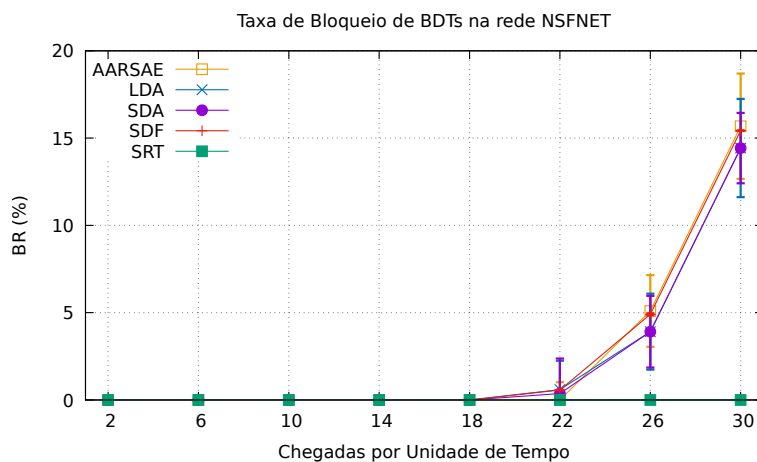


Figura 4. Taxa de bloqueios de requisições BDTs na topologia NSFNET

A menor taxa de bloqueio de requisições é obtida pela política SRT, que prioriza as requisições cujo tempo de espera é o menor. Essa estratégia aumenta as chances de que uma chamada seja atendida, podendo submetê-la várias vezes a novas tentativas de alocação até que o limite de tempo seja atingido. A associação dessa política com estratégias de alocação *first-fit* e *last-fit* é responsável por aumentar o número de canais disponíveis para o estabelecimento de requisições, dentro do cenário apresentado.

De maneira geral, os resultados das simulações mostram que a adoção de

uma estratégia de escalonamento melhora os índices de atendimento de diferentes tipos de serviços na rede. Enquanto um algoritmo que não realiza escalonamento tende a declinar o percentual de aceitação quando ocorrem a partir de 10 chegadas por unidade de tempo, com uma política de classificação esse comportamento só ocorre a partir de 18 chegadas de requisições por unidade de tempo. Além disso, a política SRT é a mais vantajosa para ambos os tipos de demandas na rede, BDT e MBDT, pois tende a postergar um possível bloqueio até o máximo limite de tempo permitido.

5. Conclusão

Este trabalho propõe um estudo das políticas de escalonamento de demandas BDTs e MBDTs em um ambiente de tráfego dinâmico em EON. Para lidar com os dois tipos de serviço, as técnicas de engenharia de tráfego selecionam os caminhos menos congestionados, e nestes, promove alocação de espectro com as políticas *first-fit* e *last-fit*, proporcionando uma separação mínima do enlace óptico por tipo de serviço. Para melhorar a taxa de atendimento de ambos os tipos de requisições, as políticas de escalonamento SDF, LDA, SDA e SRT são implementadas. Os melhores resultados foram obtidos com a política SRT, que permite adiar o máximo possível o tempo no qual a demanda deve ser atendida. Além disso, os algoritmos SDA e SDF demonstraram bom desempenho no cenário de tráfego elevado (mais de 10 chegadas por unidade de tempo), uma vez que mantém a taxa de atendimento em torno de 65% de ressincronizações bem-sucedidas, mostrando que a ideia de atender primeiro aquelas demandas de menor granularidade, tanto de volume de dados quanto de tempo para entrega no destino, melhoram a fluidez dos recursos.

Como trabalhos futuros, pretende-se estender essa proposta, baseada em IR para realizar também AR de recursos na dimensão do tempo. Além disso, outros tipos de tráfegos serão acrescentados a esse cenário para simular um ambiente mais próximo da realidade, com compartilhamento dos canais de comunicação.

Referências

- Aibin, M. and Walkowiak, K. (2017). Analysis of inter-data center elastic optical network. In *19th International Conference on Transparent Optical Networks (ICTON), 2017*, pages 1–4. IEEE.
- Castro, M. and Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461.
- Chatterjee, B. C., Kitsuwon, N., and Oki, E. (2017). Performance evaluation of first-last-exact fit spectrum allocation policy for elastic optical networks. In *2017 19th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.
- Costa, L. R., de Sousa, L. S., de Oliveira, F. R., da Silva, K. A., Júnior, P. J. S., and Drummond, A. C. (2016). Ons: Simulador de eventos discretos para redes ópticas wdm/eon. In *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. Sociedade Brasileira de Computação (SBC).

- Deng, G. and Wang, K. (2018). An application-aware qos routing algorithm for sdn-based iot networking. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00186–00191.
- Li, G., Qian, Y., Liu, L., and Yang, Y. R. (2018). Jms: Joint bandwidth allocation and flow assignment for transfers with multiple sources. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 123–130. IEEE.
- Lu, W., Liang, L., and Zhu, Z. (2017). On vnf-sc deployment and task scheduling for bulk-data transfers in inter-dc eons. In *IEEE/CIC Int. Conf. Communications in China (ICCC)*, pages 1–4.
- Lu, W. and Zhu, Z. (2015). Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks. *Journal of Lightwave Technology*, 33(10):2078–2086.
- Luo, L., Yu, H., and Ye, Z. (2018). Deadline-guaranteed point-to-multipoint bulk transfers in inter-datacenter networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- Sousa, L. S. and Drummond, A. C. (2017). Provendo multiplas transfências de dados em massa com roteamento e alocação de espectro ciente da aplicação em redes opticas elásticas. *Simpósio Brasileiro de Redes de Computadores (SBRC)*, 35(1/2017).
- Srinivasan, S. M., Truong-Huu, T., and Gurusamy, M. (2018). Deadline-aware scheduling and flexible bandwidth allocation for big-data transfers. *IEEE Access*, 6:74400–74415.
- Wang, N., Jue, J. P., Wang, X., Zhang, Q., Cankaya, H. C., and Sekiya, M. (2015). Holding-time-aware scheduling for immediate and advance reservation in elastic optical networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 5180–5185. IEEE.
- Wu, Y., Zhang, Z., Wu, C., Guo, C., Li, Z., and Lau, F. C. (2017). Orchestrating bulk data transfers across geo-distributed datacenters. *IEEE Transactions on Cloud Computing*, 5(1):112–125.
- Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716.
- Zhang, H., Chen, K., Bai, W., Han, D., Tian, C., Wang, H., Guan, H., and Zhang, M. (2015). Guaranteeing deadlines for inter-datacenter transfers. In *Proceedings of the Tenth European Conference on Computer Systems, EuroSys '15*, pages 20:1–20:14, New York, NY, USA. ACM.