Provendo Múltiplas Transferências de Dados em Massa com Roteamento e Alocação de Espectro Ciente da Aplicação em Redes Ópticas Elásticas

Léia Sousa de Sousa, André Costa Drummond

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

Abstract. *This paper presents three Application-Aware Routing and Spectrum Assignment (AARSA) for the Multiple Bulk Data Transfer (MBDT). Simulations were done in an inter-data center (IDC) network to compare the application-aware routing versus conventional routing, involving both resynchronization and backup applications. The results indicate that AA-RSA outperforms conventional RSA by providing up to* 70% *more of resynchronisations among data centers without increasing the blocking rate of backup requests.*

Resumo. *Este trabalho apresenta três soluções de roteamento e atribuição de espectro ciente da aplicação (AARSA) para Múltiplas Transferências de Dados em Massa (MBDT). Simulações foram realizadas em uma rede inter-centro de dados (ICD) para comparar o roteamento ciente da aplicação versus convencional, envolvendo-se aplicações de ressincronização e backup simultaneamente. Os resultados indicam que soluções cientes superam a convencional com ganho de até 70% no número de ressincronizações ICD efetivadas, sem impactar na taxa de bloqueio de backups.*

1. Introdução

Vários avanços em tecnologias de redes ópticas elásticas (EON) têm levado os provedores de nuvem a se prepararem para futura adoção em redes inter centros de dados (ICD) geo-distribuídas. A *Microsoft*, por exemplo, que atualmente utiliza tecnologia WDM, tem mostrado interesse em explorar o uso de *transponders* de largura de banda flexível (BVTs) e implementação de roteamento de alocação de espectro com modulação fixa (RSA), uma vez que sua infraestrutura é composta de uma variada gama de tipos de fibra ópticas, representando um desafio para o uso mais eficiente do espectro óptico [Filer et al. 2016].

A EON utiliza uma grade do espectro óptico com menores espaçamentos e transporta mais *bits* por símbolos, resultando em melhor ajuste das demandas às larguras de banda dos canais. Os BVTs da EON convertem o sinal vindo da rede cliente para um sinal a ser comutado no domínio óptico, e implementam técnicas *Orthogonal Frequency-Division Multiplexing*(OFDM) capazes de alocar fatias do espectro (*slots*) para um determinado caminho de acordo com a taxa de transmissão solicitada. Outros importantes elementos na EON são os *switches cross-connects* de largura de banda flexível (BV-WXCs), que estabelecem caminhos ópticos com os recursos solicitados por uma dada aplicação e podem suportar uma infinidade de caminhos com as mais variadas capacidades [Wan et al. 2012].

A principal preocupação dos provedores de serviços em nuvem é como lidar com o enorme volume de tráfego, que tem crescido 29% a cada ano, e como lidar com o crescente

período de tempo no qual a largura de banda permanece ocupada [Cisco 2016]. Nesse cenário, as aplicações de Transferências de Dados em Massa (BDT) e as múltiplas BDTs (MBDT), que pertencem à classe de tráfego de menor prioridade (classe *background*), acabam sendo penalizadas em detrimento de outras aplicações de maior prioridade, como as de vídeo IP. Isso ocorre porque BDTs e MBDTs movimentam volumosas quantidades de dados em redes ICD, são tolerantes a atrasos e podem esperar enquanto oportunidades são dadas ao tráfego que deve ser entregue com o menor retardo possível sob pena de arruinar a experiência dos usuários [Zhang et al. 2015].

As aplicações de transferências de dados mais comuns são: (i) os *backups*, transferências unitárias (BDT) correspondentes a um conjunto de vários tipos de dados que, por medidas de proteção, são diariamente encaminhados para CDs remotos e pertencentes a diferentes zonas de desastres (DZ), e (ii) as ressincronizações, que são MBDTs implementadas para atualizar grandes conjuntos de dados distribuídos, como forma de replicação para garantir tolerância a falhas. Essas aplicações podem ser favorecidas com a maior disponibilidade em EON, abrindo precedentes para a implantação de roteamento ciente da aplicação (*Application-Aware Routing*, AA-R), seguindo a tendência de adoção de elementos da rede de transporte capazes de distinguir fluxos e pacotes variados [Sadasivarao et al. 2016], graças a futura integração das tecnologias de rede de transporte óptica (OTN) e *rede definida por software* (SDN). O escalonamento de requisições é outra técnica que favorece o aproveitamento dos recursos da rede, permitindo que as requisições busquem por novas oportunidades de atendimento [Laoutaris et al. 2011].

Enquanto a sincronização requer que o nó detentor dos dados seja paralisado para que ocorra a transferência para todas as suas réplicas [Markowski 2016], na ressincronização segura, quando um lote de comunicação de muitos para um ($M \rightarrow 1, M \geq 3$) é estabelecido, mais nós são envolvidos no papel de emissor. Todas as operações são realizadas com os membros do grupo de replicação em franco funcionamento, e a quantidade de réplicas necessárias varia de acordo com o protocolo de replicação e o modelo de sistema adotado. Além disso, no modelo mais severo de falhas, as falhas Bizantinas, são necessárias 4 réplicas funcionando corretamente para que uma delas possa falhar [Castro and Liskov 2002]. Embora este número possa ser aumentado para permitir que mais réplicas falhem, na prática utilizam-se poucas réplicas [Vukolić 2010].

Este trabalho propõe soluções para o problema RSA, que são cientes da aplicação, e atendem aplicações de *backup* e ressincronização ICD. O principal objetivo é garantir que as ressincronizações, aplicações mais complexas, sejam efetuadas e façam uso mais eficiente dos recursos da rede, atendendo um número maior de conexões e aumentando o desempenho do sistema. Além disso, o escalonamento de chamadas será adotado para aumentar as chances de atendimento em momentos e oportunidades posteriores.

Os resultados obtidos indicam que o algoritmo ciente da aplicação tem taxa de sucesso de ressincronização até 70% maior do que com o uso de uma solução convencional. Além disso, o escalonamento de requisições permite que até 25% de todas as requisições que seriam bloqueadas por falta de banda, sejam atendidas em outros momentos, reduzindo as despesas com armazenamento em trânsito. Destaca-se ainda que, mesmo provendo ressincronizações com alto índice de sucesso, as operações de *backups* não foram penalizadas, uma vez que o percentual de bloqueio permaneceu entre 10% e 12%. Assim, as principais contribuições deste trabalho são: *(i)* proposta de três algoritmos RSA em

EON cientes da aplicação: AARSAE (sem escalonamento), MR (com escalonamento das ressincronizações) e MR+B (com escalonamento das ressincronizações e *backups*); e *(ii)* realização de simulações para avaliar o desempenho dos algoritmos propostos em comparação com uma solução RSA convencional, demonstrando claramente os ganhos advindos da exploração de informações das aplicações na rede.

2. Trabalhos Relacionados

Muitos dos grandes provedores de serviços em nuvem pretendem migrar suas tecnologias de rede de transporte atuais para a EON [Filer et al. 2016], motivados pela promessa de aumento nas taxas de transmissão de dados e utilização mais eficiente do espectro óptico [Wan et al. 2012]. Mas os benefícios irão além disso. As aplicações que são executadas nessa infraestrutura e os diferentes tipos de dados que nela transitam poderão ser identificados com especificidades pelos dispositivos da rede [Sadasivarao et al. 2016], e as que mais demandam recursos e representam custos significativos nas despesas dos CDs poderão ser beneficiadas com um roteamento mais inteligente, o AA-R. Outro tipo de roteamento inteligente é o roteamento ciente das limitações na camada física, que já é um tema tradicionalmente tratado em redes ópticas [Subramaniam et al. 2013].

Atualmente, as melhores soluções para lidar com o tráfego BDT e MBDT relativo a aplicações como *backups* e ressincronizações são baseadas em roteamento estático, envolvendo operações de escalonamento em diferentes janelas de tempo [Nandagopal and Puttaswamy 2012], mesmo quando canais comerciais de ISPs são utilizados de maneira associada à redes dedicadas [Li et al. 2012] e ainda, utilizando a largura de banda ociosa em qualquer ponto da rede, bem como apoiando-se em previsões de consumo de banda seguindo o histórico da rede [Laoutaris et al. 2011].

A partir do surgimento da EON, o atendimento de requisições orientadas a dados é tido como uma das soluções para aproveitar a fragmentação do espectro óptico [Lu et al. 2015b, Lu and Zhu 2015], no entanto, as aplicações dessa natureza ainda precisariam esperar por recurso suficiente, o que representaria fragmentações de tamanho considerável, ou continuariam alugando serviços de armazenamento ao longo da rede, aumentando o grau de vulnerabilidade dos dados. Por outro lado, [Wan et al. 2012] propõe duas soluções RSA convencionais, uma delas com modulação fixa e outra com modulação adaptativa a partir do nível mais eficiente para satisfazer a mais demandas.

Soluções de AA focadas em EON têm começado a ser exploradas na literatura, a exemplo de [Song et al. 2012], que propõe o mapeamento de máquinas virtuais aos seus respectivos *hosts* físicos para realizar migrações de acordo com a disponibilidade de recursos e com as distâncias entre esses *hosts*. Como resultado, reduziu-se o tráfego de dados nos enlaces da rede e o consumo de energia dentro do CD. Esse tema também tem começado a ser destacado no campo das SDNs [Zinner et al. 2014]. Em [Sousa et al. 2016] são mostradas as vantagens do roteamento ciente da aplicação em EON, comparada com a uma solução RSA convencional. A principal delas é o aumento significativo de requisições de grandes demandas sendo servidas dinamicamente.

Este trabalho também propõe três soluções cientes da aplicação, em um cenário com duas aplicações distintas concorrendo por recursos. Por serem críticas, o objetivo principal é manter o funcionamento bem sucedido das ressincronizações, sem elevar a taxa de bloqueio de banda para *backups*.

3. Problema das Transferências de Dados em Massa para as aplicações de Backup e Ressincronização ICD

A computação em nuvem tem permitido que mais dados sejam produzidos por várias gamas de aplicações. Assim, os requisitos de alta disponibilidade, redundância e proteção contra falhas são partes fundamentais das garantias de confiabilidade e funcionamento contínuo dos serviços. Operações de missão crítica como *backups* e ressincronização são implementadas para fornecer disponibilidade e proteção contra falhas [Agrawal et al. 2013].

Para tolerar uma maior variedade de falhas, conjuntos de réplicas são implantadas de maneira geograficamente distribuída. Cada CD da rede armazena uma partição de dados, que é replicado separadamente. Um grupo de replicação é um grupo de CDs responsável pela mesma partição. Assim, cada CD pode participar simultaneamente de vários grupos diferentes. O Sistema de Gerenciamento de Dados Distribuídos (SGDD) que lida com esses CDs possue um correto e consistente mapeamento dos grupos de replicação em seus membros e suas respectivas localizações [Sharov et al. 2015].

Em cada um dos grupos de replicação, a comunicação entre seus integrantes ocorre por meio de várias rodadas de trocas de mensagens. Existem variados protocolos que controlam essa comunicação com o objetivo de sempre alcançar um estado comum entre as réplicas de dados. Quando um estado desejável e comum é alcançado, diz-se que as réplicas estão síncronas [Castro and Liskov 2002]. Um CD inativo, que deseja voltar à rede após um período de indisponibilidade e cujo estado encontra-se desatualizado, pode submeter uma mensagem solicitando integrar-se a um dado grupo e assim, acionar o serviço de ressincronização. O grupo de réplicas designado pelo SGDD possui o mesmo estado e um mapeamento das partições a serem replicadas [Agrawal et al. 2013].

A ressincronização ocorre por meio de MBDT originadas de CDs definidos pelo SGDD e recebidas pelo CD desatualizado, ocorrendo dentro de um período de tempo suficiente para a completa atualização do CD solicitante e consumindo recursos na rede. Enquanto *backups* realizam operações individuais de BDT, as ressincronização realizam MBDT de dados relacionados. Ambas são de mesma natureza, pois pertencem à classe de tráfego (*background*) tolerante a atrasos e requererem grandes quantidades de banda [Lu et al. 2015a]. A camada de rede atende várias solicitações de aplicações de diferentes classes de tráfego de forma transparente com roteamento convencional.

As soluções de roteamento ciente da ressincronização ICDs que serão apresentadas, possuem maior complexidade devido a tomada de decisão sobre uma requisição MBDT, feita através da busca de combinações de sub-requisições aptas à transferir partições de dados. Dessa maneira, como qualquer subconjunto de requisições é capaz de realizar uma ressincronização bem sucedida, e atender os requisitos de tolerância a falhas Bizantinas, as demais requisições são redundantes. Para resolver esse problema, além de tentar estabelecer caminhos ópticos com espectro suficiente, essas tentativas são experimentadas para todas os subconjuntos de 3 CDs transmissores que integram o grupo de replicação [Vukolić 2010].

Para prover recursos às BDTs e MBDTs mediante AARSA, a rede foi modelada como um grafo direcionado $G(V, E)$, com V e E representando o conjunto de BW-WXCs e enlaces da fibra, respectivamente. Assume-se que alguns $v \in V$ conectam-se diretamente a CDs, representados como V^{CD}, hábeis a originar e receber transferências de dados. Cada enlace $e \in E$ pode acomodar no máximo, BW *slots* de frequência do espectro.

Requisição BDT. As requisições BDTs são representadas 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, $\beta^r_{min} = \frac{C_r}{dl_r}$, ou taxa máxima, β^r_{max}, equivalente a capacidade de banda disponível no canal no momento em que a solicitação é atendida.

Escalonamento de requisições BDTs. Quando uma requisição não pode ser atendida imediatamente, ela é escalonada em uma janela W para novas tentativas do serviço. Na janela, o tempo restante (t^{rem}) para atendimento é equivalente ao tempo atual (t^{now}) subtraído do prazo total (dl_r), ou seja, $t^{rem}_w = dl_r - t^{now}$. A taxa mínima para transferir C_r dentro do período de tempo t^{rem}_w é $\beta^r_{mim_w}$, isto é, $\beta^r_{mim_w} = \frac{C_r}{t^{rem}_w}$, com $\beta^r_{mim} < \beta^r_{mim_w}$.

Requisição MBDT. Uma requisição MBDT é um conjunto de requisições BDT, que é representada como $R = \{r_1, r_2, ..., r_n\}$ onde $dl_{r_i} = dl_{r_2} = ... = dl_{r_n}$. A ressincronização ocorre sobre uma única partição de dados, assim as sub-requisições possuem pequena proximidade de tempo. Para efetiva ressincronização e devido ao limite de tolerância a falhas Bizantinas [Castro and Liskov 2002], três sub-requisições devem ser atendidas. No caso de requisições MBDT com mais do que três requisições de transferência, o algoritmo considera que a aplicação apenas requer o mínimo de três BDTs. Atender a mais do que três sub-requisições leva ao desperdício de recursos. A taxa mínima a ser atribuída para R é definida como β^R_{min}, que é equivalente ao mínimo de banda tal que um número suficiente de $r \in R$ sejam atendidas, e a taxa máxima, β^R_{max}, é equivalente a máxima capacidade de banda disponível no canal no momento do atendimento.

Escalonamento de MBDT. Analogamente ao escalonamento de BDT, quando uma requisição MBDT não pode se servida em uma dada tentativa de busca por recursos, essa requisição é escalonada e espera uma nova oportunidade na janela W. Como o prazo para a transferência diminui, a largura de banda para cada $r \in R$ deve ser atualizada. A taxa mínima para R dentro da janela é $\beta^{R_w}_{min_w} = \sum_1^n \beta^{r_w}_{min_w}$, sendo $\beta^R_{mim} < \beta^{R_w}_{mim_w}$.

As taxas máximas e mínimas para atender BDT podem ser representadas genericamente por $taxa(r)$, e da mesma forma, para atender MBDT, as taxas podem ser generalizadas como $taxa(R)$. Essa notação será utilizada à seguir.

4. Provendo Backups e Ressincronizações

As soluções cientes da aplicação que serão apresentas proveem serviços de *backups* e ressincronização. Serão mostrados os seguintes algoritmos: *(i)* Algoritmo de Roteamento e Alocação de Espectro Ciente da Aplicação Estendido, AARSAE (Algoritmo 1), que atenderá requisições BDT e MBDT; *(ii)* Algoritmo do Máximo de Ressincronizações (MR); e, *(iii)* Algoritmo do Máximo de Ressincronizações e *Backups* (MR+B). Os algoritmos apresentados em (*ii*) e (*iii*) podem invocar as soluções genéricas *ServeBDT* e *ServeMBDT* para a primeira tentativa de servir uma requisição BDT e MBDT, e no caso de requisições serem escalonadas, podem invocar os algoritmos ServeMBDTinW (Algoritmo 2) e ServeRequisicaoInW (Algortimo 3).

4.1. Roteamento e Alocação de Espectro Ciente da Aplicação Estendido (AARSAE)

O AARSAE (Algoritmo 1) atende BDTs e MBDT. Para atender as MBDT o AA-RSA [Sousa et al. 2016] é invocado na linha 2. Para atender as requisições BDT, os

procedimentos das linhas 4-13 são utilizados, começando com a busca de caminhos candidatos pelo $KSP(r, K)$ [Yen 1971]. Em cada caminho verifica-se se a taxa mínima está disponível (linha 6). Em caso positivo, as restrições RSA são testadas (linha 7) para que r seja aceita(linha 8). Caso contrário r é bloqueada (linha 10).

Algoritmo 1 AARSAE(r, R)

```
1:  if requisição é R then                                  ▷ Requisição MBDT
2:      AA-RSA(R)
3:  else                                                    ▷ Requisição BDT
4:      KSP(r, K)
5:      for k = 1 → K do
6:          if tem β^r_min em k then
7:              if serve restrições RSA then
8:                  Aceita (r, k, β^r_min)
9:              else
10:                 Bloqueia r
11:             end if
12:         end if
13:     end for
14: end if
```

4.2. Soluções com Escalonamento

O algoritmo anterior, AARSAE, não faz escalonamento de requisições. Considera-se agora o uso de escalonamento para viabilizar novas reconfigurações de recursos. Ao consumir quase toda a largura de banda disponível, alguns pedidos seriam bloqueados. Com o escalonamento, em vez de bloquear uma requisição devido à falta de recursos, essa requisição é colocada em uma janela W, supervisionada pelo plano de controle da rede, que verifica se o prazo para uma determinada solicitação está se esgotando e é necessário alocar o máximo de banda suficiente para atendê-la. As reconfigurações ocorrem nas chegadas e partidas de requisições, e quando o tempo de uma solicitação na janela atinge o limite.

4.2.1. Soluções Genéricas

Existem dois algoritmos definidos como Soluções Genéricas: *ServeBDT* e ServeMBDT. O ServeBDT é similar aos procedimentos das linhas 4-13 do algoritmo AARSAE (Algoritmo 1), exceto por uma razão: a *taxa* a ser atribuída para atender BDT também é um parâmetro de entrada, $(r, taxa(r))$, e pode ser definida pelo algoritmo que o invocar. O algoritmo ServeMBDT é similar ao AA-RSA [Sousa et al. 2016], exceto por duas diferenças básicas: *(i)* os parâmetros de entrada para ServeMBDT são $(R, taxa(R))$; e *(ii)* a $taxa(R)$ não é atribuída em função de comparações com o diâmetro da rede, conforme o AA-RSA [Sousa et al. 2016], mas sim, é definida pelo algoritmo que o invocar.

4.2.2. Reconfiguração Pós-Escalonamento

Se uma requisição não pode ser servida na primeira tentativa, ela é encaminhada para uma janela de espera W. Em W existem dois procedimentos definidos, o *ServeMBDTinW* (Algoritmo 2) que escalona ressincronizações, e o *ServeRequisicaoInW* (Algoritmo 3), que escalona *backups* e ressincronizações, atendendo as ressincronizações com maior

prioridade. Ambos os algoritmos invocam as rotinas *ServeBDT* e *ServeMBDT* (Soluções Genéricas), passando como parâmetro o tipo de chamada e taxa solicitada por tal chamada.

O algoritmo *ServeMBDTinW* (Algoritmo 2) coloca requisições MBDT na fila $\mathscr{F}^R$ ordenada pelo prazo (linha 1). Para cada elemento nessa fila (linha 3) verifica-se se taxa solicitada é mínima (linhas 4-6) ou máxima (linha 8). A taxa mínima equivale a soma de todas as taxas mínimas de sub-requisições, calculadas a partir do tempo restante para atendimento (linha 6). A taxa máxima equivale a máxima disponibilidade na rede. O algoritmo *ServeMBDT* é requisitado na linha 12. Como a janela contendo as requisições MBDT é monitorada pelo plano de controle, se alguma chamada estiver alcançando o prazo final de atendimento antes de ser definitivamente bloqueada, o algoritmo *ServeMBDT* é requisitado diretamente para esta chamada.

Algoritmo 2 ServeMBDTinW($R, taxa(R)$)

```
1:  F^R ← R
2:  for R_w ← 1 to |F^R| do
3:      for r_w ← 1 to |R_w| do
4:          if taxa(R) = β^R_min then
5:              t^rem_w = dl_r − t^now
6:              taxa(r)_w = (C/t^rem_w)
7:          else
8:              taxa(R)_w ←Máximo disponível
9:          end if
10:         taxa(R)_w ← Max(taxa(r)_w)
11:     end for
12:     ServeMBDT(R_w, taxa(R)_w)
13: end for
```

O algoritmo *ServeRequisicaoInW* (Algoritmo 3) trata requisições MBDT e BDT enfileiradas. Para atender as MBDT ele invoca o algoritmo *ServeMBDTinW*, linha 5. Para atender as BDTs da fila, primeiramente verifica-se se não existem MBDTs esperando (linha 7). Em seguida, processa-se a taxa solicitada, que pode ser a mínima, considerando-se o prazo restante (linhas 9-11), ou pode ser a máxima disponível no canal (linha 13). A taxa definida e a requisição BDT em questão são passadas ao algoritmo *ServeBDT* na linha 15. Alertas também são emitidos pelo plano de controle para atender MBDT ou BDT que estejam próximas de encerrar o prazo de atendimento.

4.2.3. Máximo de Ressincronizações (MR)

As soluções prévias possuem os parâmetros genéricos $taxa(r)$ e $taxa(R)$ para que diferentes taxas possam ser solicitadas para atender r e R, sejam taxas mínimas ou máximas. No algoritmo MR (Algoritmo 4), que atende r (linhas 3-4) e R (linhas 6-10), as taxas solicitadas equivalem à taxa máxima disponíveis no canal no momento da solicitação. Assim, um grande número de requisições pode ser admitido, desocupando recursos tão rápido quanto possível. O algoritmo MR faz escalonamento de MBDT, o que significa que após a primeira tentativa de atendimento de uma chamada R (linha 6), se for mal sucedida (linha 7), todas as requisições desse tipo são encaminhadas para uma fila de espera, com a invocação do algoritmo *ServeMBDTinW* (linha 8), e podem ser submetidas a qualquer momento à novas reconfigurações, enquanto o prazo total não tiver transcorrido. Por esse

Algoritmo 3 ServeRequisicaoInW($(r, taxa(r))$,$(R, taxa(R))$)

```
1: 𝓕 ← R ∪ r
2: Ordena 𝓕 priorizando R
3: for f ← 1 to |𝓕| do
4:     if f = R then
5:         ServeMBDTinW(R, taxa(R))
6:     else
7:         if ¬∃R_w(R_w ∈ 𝓕) then
8:             for r_w ← 1 to |𝓕| do
9:                 if taxa(r) = β^r_min then
10:                    t^rem_w = dl − t^now
11:                    taxa(r)_w = (C/t^rem_w)
12:                else
13:                    taxa(r)_w ← Máximo disponível
14:                end if
15:                ServesBDT(r_w, taxa(r)_w)
16:            end for
17:        end if
18:    end if
19: end for
```

motivo, a janela W sempre é verificada. As BDTs não são escalonadas e seu atendimento é feito invocando a rotina *ServeBDT*. Caso a requisição não seja atendida, ela é bloqueada.

Algorithm 4 MR(G, r, R)

```
1: Verifica W
2: for i ← 1 to (Σ(R) + Σ(r)) do
3:     if i = r then
4:         ServeBDT(r, β^r_max)
5:     else
6:         ServeMBDT(R, β^R_max)
7:         if Se R não for atendida then
8:             ServeMBDTinW(R, β^R_max)
9:         end if
10:    end if
11: end for
```

4.2.4. Máximo de Ressincronizações e *Backups* (MR+B)

No algoritmo MR+B (Algoritmo 5), as rotinas *ServeBDT* e *ServeMBDT* são invocadas, cada uma para atender sua respectiva chamada (r e R). Ao tentar atender BDT (linha 4), se a chamada não puder ser atendida (linha 5), ela é imediatamente escalonada e o algoritmo *ServeRequisicaoInW* (linha 6) faz novas tentativas. De maneira similar, as chamadas MBDT tentam ser servidas pelo algoritmo *ServeMBDT* (linha 9), e caso a tentativa falhe (linha 10), essa chamada pode ser escalonada para esperar por outras tentativas de atendimento quando é invocado o algoritmo *ServeRequisicaoInW* (linha 11).

Em todas as oportunidades, as BDTs solicitam a taxa mínima, inclusive quando o algoritmo *ServeRequisicaoInW* é invocado (linha 5). O algoritmo MR+B atribui maior prioridade à MBDT devido a sua maior complexidade e necessidade de banda. A ideia é que a taxa de bloqueio de BDT não cresça e ainda assim, que estas não saturem os recursos disponíveis para que as ressincronizações continuem sendo atendidas.

Algoritmo 5 MR+B(G, r, R)

1: Verifica W
2: **for** $i \leftarrow 1$ to $(\Sigma(r) + \Sigma(R))$ **do**
3: **if** $i = r$ **and** $ServesBDT(r, \beta^r_{mim})$ não atende r **then**
4: $ServeRequisicaoInW(i, taxa(i))$, $taxa(i) = \beta^r_{mim}$
5: **end if**
6: **if** $i = R$ **and** $ServeMBDT(R, \beta^R_{max})$ não atende R **then**
7: $ServeRequisicaoInW(i, taxa(i))$, $taxa(i) = \beta^R_{max}$
8: **end if**
9: **end for**

Complexidade dos Algoritmos

O AARSAE (Algoritmo 1) lida com requisições BDTs e MBDTs. Para atender MBDT, o algoritmo AA-RSA [Sousa et al. 2016] é invocado, e sua complexidade de tempo é $O\left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right)$. O serviço para BDT apenas requer a busca por caminhos feita pelo KSP e política de atribuição *First-Fit*. Assim, a complexidade de tempo total para esse algoritmo é de $O\left(\left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right) + K^3V^3\right)$.

Os algoritmos ServeBDT e ServeMBDT, que são Soluções Genéricas (Subseção 4.2.1), lidam com BDT e MBDT, respectivamente. A complexidade de tempo do Algoritmo *ServeBDT* é $O(K^3V^3)$, e a do Algoritmo *ServeMBDT* é $O\left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right)$.

Os Algoritmos 2 e 3 escalonam requisições e por isso realizam reconfiguração pós-escalonamento (Subseção 4.2.2) . Existe uma fila de requisições e operações são definidas para atualizar o tempo restante para a transmissão, do qual se deduz a taxa. Essas operações são de complexidade de tempo linear. A atualização para requisições na fila, no pior caso pode ser feita para todas as requisições encaminhadas, respectivamente. Assim, a complexidade de tempo do Algoritmo 2 é $O\left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right)$ e do Algoritmo 3 é $O\left(\log n\left[(K^3V^3) + \left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right)\right]\right)$.

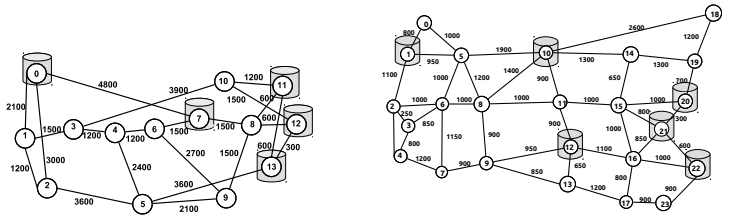
Soluções genéricas e reconfiguração pós-escalonamento são invocadas pelos algoritmos MR e MR+B. Assim, a complexidade de tempo do algoritmo MR é $O\left(K^3V^3 + \left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right)^2\right)$ e a do algoritmo MR+B é $O\left(\log n\left[(K^3V^3)^2 + \left(\left(\frac{n!}{b!(n-b)!}\right) * V^3\right)^2\right]\right)$.

Embora os algoritmos cientes da aplicação apresentem complexidade de tempo fatorial devido às combinações sobre MBDT, a literatura mostra que o tamanho de b geralmente é de 3 requisições de ressincronizações, visto que no mundo real esse é o tamanho mais comum dos conjuntos de réplicas de centros de dados, por ser desvantajoso,

do ponto de vista do custo capital e operacional, possuir um conglomerado muito grande de recursos que são poucos solicitados [Vukolić 2010].

5. 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 1(a)) e USA (Figura 1(b)) para comparar o desempenho com o algoritmo RSA [Wan et al. 2012]. Em cada figura abaixo os nós são WXCs, as arestas estão numeradas com as respectivas distâncias e tem-se alguns dos nós destacados por terem conexão direta com CDs que efetuam ou recebem uma transferência de dados.



(a) (b)

Figura 1. Topologias de rede (a) NSFNET e (b) USA.

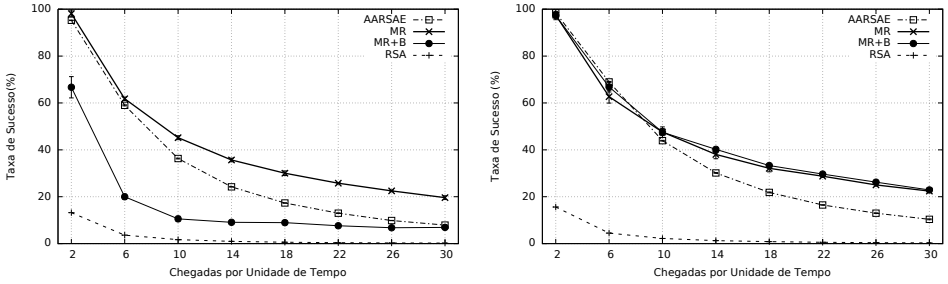
Considerou-se o uso de 15 BVTs em cada nó, sendo cada um com capacidade de transmissão de 8 *slots*. Cada *slot* tem largura de banda de $12.5GHz$ e cada enlace possui 120 *slots* de frequência com espectro total de $1.5THz$. Para banda de guarda assumiu-se dois *slots*. Os algoritmos foram testados usando a modulação *Quadrature Phase Shift Keyed* (QPSK).

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, 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. As BDTs foram configuradas para transferir $100GB$ e $300GB$ dentro de um prazo de 20 unidades de tempo. Para as MBDTs foram definidas um total de 4 requisições para um mesmo destino, com volume de dados de $100GB$ e $500GB$ para serem transferidos dentro de um prazo de 100 unidades de tempo.

Taxa de Sucesso das Ressincronizações (MBDT)

A taxa de sucesso da ressincronização é calculada como o número de chamadas MBDT aceitas, dividido pelo total de requisições do mesmo tipo. O roteamento convencional não lida com requisições agrupadas (conjunto de requisições relacionadas), assim, precisa servir individualmente e independentemente cada chamada. Já o roteamento ciente da aplicação, lida com conjuntos de requisições relacionadas fazendo combinações em vários subconjuntos com três requisições a partir do conjunto R e selecionando um subconjunto. Isso significa que, em conjuntos com 4 requisições, uma delas será descartada.

A Figura 2(a) mostra os resultados para os algoritmos cientes da aplicação na rede NSFNET. O algoritmo RSA [Wan et al. 2012] tem desempenho inferior aos demais por atender chamadas individualmente e não são direcionadas a atualizar a mesma partição de dados, ao contrário dos algoritmos que são cientes da aplicação. Para atender qualquer tipo de chamada, a taxa mínima é atribuída, tornando os canais ocupados por períodos mais longos. Com o aumento do número de chegadas, as tentativas de satisfazer as solicitações de um conjunto falham por falta de banda disponível, e o esgotamento do prazo também influencia nesse resultado. Sua taxa de sucesso começa em 13.2% para cargas mais baixas, e cai para 0.2% com a saturação dos recursos.



(a) (b)

Figura 2. Taxa de sucesso de ressincronizações nas redes (a) NSFNET e (b) USA.

O algoritmo MR+B serve as ressincronizações com a taxa máxima, resultando em 66% de atendimento com carga mais baixa. Tanto BDT quando MBDT são escalonadas, mas a prioridade de atendimento é dada às MBDTs que, atendidas com taxa máxima, desocupam a banda utilizada mais rapidamente. Muitos BDTs são estabelecidos e permanecem ativos por longos períodos, por receberem taxa mínima, levando os recursos da rede à exaustão, garantindo taxa sucesso de ressincronizações entre 95% (com carga baixa) e $7,9\%$ (com carga alta) entre todas as requisições desse tipo.

O melhor resultado é obtido pelo algoritmo MR que serve BDT e MBDT com taxas máximas e mantém disponibilidade de banda por mais tempo por desocupá-la com rapidez, alcançando de 98% e 19% de sucesso de MBDT. Quando a carga na rede é baixa, quase 25% de sub-requisições são eliminadas ou descartadas das MBDT. Com carga alta, essa taxa cai para 5%. Ambos MR (Algoritmo 4) e MR+B (Algoritmo 5) escalonam requisições MBDT. Graças às várias oportunidades de atendimento oferecidas a cada chamada, em média 43% (para MR) e 50% (para o MR+B) dessas requisições foram atendidas após a primeira tentativa de serviço. Como os recursos se tornam mais escassos, o escalonamento se mostra vantajoso para as requisições com maiores requisitos de banda.

Os resultados na rede USA (Figura 2(b)) sugerem que em uma rede bem conectada, priorizar as MBDT e oferecê-las a taxa máxima resulta em até 98% de ressincronizações bem sucedidas, quando a carga é baixa. Por esta razão, os algoritmos AARSAE $(98,4\%)$, MR $(97,4\%)$ e MR+B $(97,2\%)$ alcançaram bons resultados. O descarte de requisições pelo algoritmo MR é de até $24,6\%$, enquanto o algoritmo MR+B elimina $24,5\%$ e o AARSAE, $24,7\%$. Cerca de 42% das ressincronizações bem sucedidas dos MR e MR+B ocorrem depois que as MBDT são escalonadas.

Em geral, as soluções cientes da aplicação podem ser favorecidas quando diferentes

taxas são alocadas e com a combinação de sub-requisições a partir de uma requisição MBDT, que descarta uma solicitação desnecessária de cada MBDT. Com isso economizou-se largura de banda com todas as soluções cientes da aplicação.

Taxa de Bloqueio de Requisições de *Backups* (BR)

A taxa de bloqueio (BR) de *Backups* (BDT) equivale ao número de requisições r bloqueadas, divididas pelo total de requisições desse mesmo tipo. O objetivo das soluções propostas é manter taxa de aceitação das MBDTs em um cenário com mais aplicações executando. No entanto, é desejável que, com o aumento na taxa de aceitação de requisições MBDT, as demais aplicações não sejam penalizadas com bloqueio elevado.

A Figura 3(a) compara a BR de requisições BDTs quando o número de chegadas aumenta de 2 para 30 chegadas de requisições por unidade de tempo na rede NSFNET. O RSA [Wan et al. 2012] possui elevada BR, variando entre $7,5\%$ e 16%, devido às MBDTs que também requisitam serviço e ocupam a banda por mais tempo.

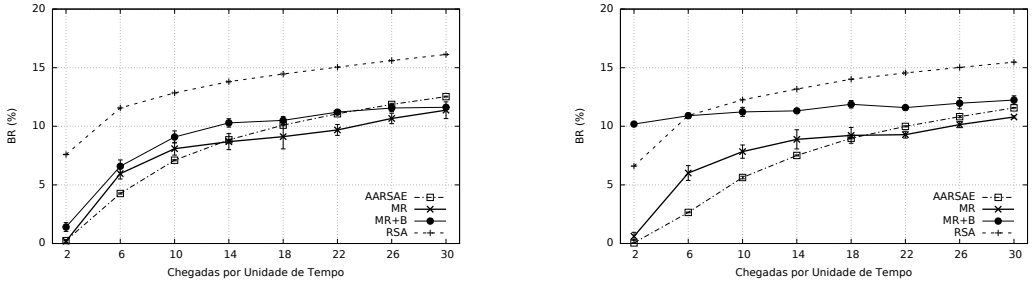


Figura 3. Probabilidade de bloqueio de BDTs nas redes (a) NSFNET e (b) USA.

O algoritmo MR+B, por outro lado, escalona os dois tipos de requisições existentes, BDTs e MBDTs, o que contribui para uma menor BR, que permanece entre $1,4\%$ e $11,6\%$. Os algoritmos AARSAE (Algoritmo 1) e MR não escalonam requisições BDTs e proveem a mínima e máxima largura de banda disponível, respectivamente, para esses *backups*, mostrando dessa maneira que, quanto mais se pode alocar, menor é a taxa de bloqueio. Adicionalmente, para servir MBDT , o AARSAE define a taxa com base na comparação do tamanho dos caminhos candidatos à metade do diâmetro da rede, passando por grandes variações de disponibilidade de recursos para atender BDTs. O MR sempre escalona MBDTs atribuindo-lhes taxa máxima, contribuindo para reduzidas BR de *backups*.

Na rede USA (Figura 3(b)), os algoritmos que fazem escalonamento tem um padrão diferenciado de comportamento em comparação com as demais topologias. O algoritmo MR+B escalona BDT e MBDT, entretanto, a prioridade atribuída à MBDT em detrimento das BDTs não afetam o seu resultado, visto que em média de 73% de todas as requisições BDTs são aceitas antes de necessitarem do escalonamento. Entretanto, o algoritmo MR+B mantém a taxa de bloqueio entre 10% e 12%.

O algoritmo MR escalona apenas MBDT. Como essas requisições MBDT são sempre atendidas com a taxa máxima, a indisponibilidade de banda acaba impactando no bloqueio de BDTs, com um aumento de 10% até a ocorrência de 18 chegadas de requisições por unidade de tempo, ou seja, quando a carga na rede começa a crescer. O algoritmo AARSAE tem a mais alta variação de crescimento na taxa de bloqueio, que

vai de $0,05\%$ para mais de 11%. Sua política de comparar os caminhos candidatos ao diâmetro da rede para definir a taxa para uma requisição provê bons resultados para as MBDT, mas satura rapidamente os recursos impedindo que as BDTs sejam atendidas.

Em todas as topologias, os algoritmos cientes da aplicação obtiveram menor bloqueio de BDTs em comparação com o algoritmo convencional. Na rede menos conectada (NSFNET), o escalonamento de BDTs reduz a taxa de bloqueio, devido a maior probabilidade de liberação de banda. Na rede mais conectada (USA), o atendimento mais rápido de requisições que demandam muita largura de banda resultou em carga efetiva reduzida na rede. Todavia, essa diminuição na disponibilidade não afetou o serviço de *backups*.

6. Conclusão

Este estudo comparou soluções de roteamento cientes da aplicação e soluções convencionais em EON, executando aplicações BDT e MBDT. Os resultados mostram que, se a solução é ciente da aplicação, o percentual de ressincronizações efetuadas com sucesso pode chegar a 70% se comparada a uma solução convencional. Esse resultado é alcançado sem aumentar a taxa de bloqueio, o que não é possível com o roteamento convencional, cuja taxa de bloqueio foi 6% maior. Além disso, as soluções apresentadas mostram-se mais eficientes em termos de uso de espectro e BVTs em todas as topologias de rede analisadas.

Referências

Agrawal, D., El Abbadi, A., Mahmoud, H., Nawab, F., and Salem, K. (2013). Managing geo-replicated data in multi-datacenters. In Madaan, A., Kikuchi, S., and Bhalla, S., editors, *Databases in Networked Information Systems*, volume 7813 of *Lecture Notes in Computer Science*, pages 23–43. Springer Berlin Heidelberg.

Castro, M. and Liskov, B. (2002). Practical Byzantine fault-tolerance and proactive recovery. *ACM Transactions on Computer Systems*, 20(4):398–461.

Cisco (2016). Cisco vni forecast and methodology, 2015-2020. `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html`. Acessed em 10/11/2016.

Costa, L. R., 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 e EON. In *SBRC 2016 - Salão de Ferramentas*, Salvador, Bahia.

Filer, M., Gaudette, J., Ghobadi, M., Mahajan, R., Issenhuth, T., Klinkers, B., and Cox, J. (2016). Elastic optical networking in the microsoft cloud. *Journal of Optical Communications and Networking*, 8(7):A45–A54.

Laoutaris, N., Sirivianos, M., Yang, X., and Rodriguez, P. (2011). Inter-datacenter bulk transfers with netstitcher. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 74–85. ACM.

Li, Y., Wang, H., Zhang, P., Dong, J., and Cheng, S. (2012). D4d: Inter-datacenter bulk transfers with isp friendliness. In *Cluster Computing (CLUSTER), 2012 IEEE International Conference on*, pages 597–600. IEEE.

Lu, P., Zhang, L., Liu, X., Yao, J., and Zhu, Z. (2015a). Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks. *IEEE Network*, 29(5):36–42.

Lu, W. and Zhu, Z. (2015). Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks. *Lightwave Technology, Journal of*, 33(10):2078–2086.

Lu, W., Zhu, Z., and Mukherjee, B. (2015b). Data-oriented malleable reservation to revitalize spectrum fragments in elastic optical networks. In *Optical Fiber Communications Conference and Exhibition (OFC), 2015*, pages 1–3.

Markowski, M. (2016). Utilization balancing algorithms for dynamic multicast scheduling problem in eon. *International Journal of Electronics and Telecommunications*, 62(4):363–370.

Nandagopal, T. and Puttaswamy, K. P. (2012). Lowering inter-datacenter bandwidth costs via bulk data scheduling. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, pages 244–251. IEEE.

Sadasivarao, A., Naik, D., Liou, C., Syed, S., and Sharma, A. (2016). Demystifying sdn for optical transport networks: Real-world deployments and insights. *IEEE GLOBECOM 2016*.

Sharov, A., Shraer, A., Merchant, A., and Stokely, M. (2015). Automatic reconfiguration of distributed storage. In *Autonomic Computing (ICAC), 2015 IEEE International Conference on*, pages 133–134. IEEE.

Song, F., Huang, D., Zhou, H., and You, I. (2012). Application-aware virtual machine placement in data centers. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 191–196. IEEE.

Sousa, L. S., Costa, L., Rodopoulos, F., Drummond, A., and Alchieri, E. (2016). Roteamento e Alocação de Espectro Ciente da Aplicação em Redes Ópticas Elásticas. In *SBRC 2016 - Trilha Principal*, Salvador, Bahia.

Subramaniam, S., Brandt-Pearce, M., Demeester, P., and Saradhi, C. V. (2013). *Cross-layer design in optical networks*. Springer.

Vukolić, M. (2010). The byzantine empire in the intercloud. *ACM SIGACT News*, 41(3):105–111.

Wan, X., Hua, N., and Zheng, X. (2012). Dynamic routing and spectrum assignment in spectrum-flexible transparent optical networks. *Journal of Optical Communications and Networking*, 4(8):603–613.

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.

Zinner, T., Jarschel, M., Blenk, A., Wamser, F., and Kellerer, W. (2014). Dynamic application-aware resource management using software-defined networking: Implementation prospects and challenges. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–6. IEEE.