



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**Transição de redes legadas para SDN - uma análise  
sobre o impacto do tempo de implantação, número e  
localização dos controladores**

Diogo F. T. Pontes

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientador

Prof. Dr. Marcelo Antonio Marotta

Brasília  
2021



# Formato

Este Trabalho de Graduação está seguindo o novo formato de artigo científico aprovado pelo Departamento de Ciência da Computação da Universidade de Brasília no processo SEI 23106.034780/2021-41, ata 447, documento SEI 6566503.

# Dedicatória

Dedico esse trabalho ao meu pai, minha mãe, minha irmã e minha namorada, que sempre me apoiaram e incentivaram.

# Agradecimentos

Agradeço a todo o corpo docente do departamento de Ciência da Computação da Universidade de Brasília que contribuiu para a minha formação e possibilitou a execução desse trabalho. Especialmente ao meu orientador Marcelo Marotta (CIC/UnB), que me apoiou de uma forma incrível e exerceu seu papel de orientador com excelência, e os outros professores que colaboraram com bastante empenho e dedicação e foram fundamentais nesse trabalho: Marcos Caetano (CIC/UnB), Geraldo Filho (CIC/UnB) e Lisandro Granville (INF/UFRGS).

# Resumo

SDN surgiu como um paradigma de rede alternativo separando o plano de controle do plano de dados. Em cenários do mundo real, a transição de uma rede tradicional para SDN geralmente é feita em etapas incrementais, devido aos custos e à burocracia de aquisição e implantação de novos hardwares nas empresas. Esta transição envolve a implantação de *switches* compatíveis com SDN e hardware para executar o software responsável por gerenciar o plano de controle, ou seja, o controlador. Muitos estudos sobre como posicionar e reduzir adequadamente o número de controladores, conhecido como *Controller Placement Problem* (CPP), foram realizados, mas apenas alguns deles exploraram o CPP em redes híbridas (tradicionais/SDN). Em redes híbridas, identificamos uma lacuna no número mínimo de controladores necessários para que uma SDN híbrida opere, considerando um número diferente de etapas de transição e a heurística para colocar esses controladores na rede. Portanto, nesta monografia, propomos três abordagens heurísticas baseadas em invariante de grafos para escolher o nó no qual será colocado um novo controlador, reduzindo seu número em uma rede que começa legada, torna-se híbrida e termina totalmente habilitada para SDN no final de uma transição finita em um horizonte de planejamento. As heurísticas propostas são comparadas com um modelo otimizado e têm se mostrado úteis em cenários onde o administrador da rede não tem controle total sobre a evolução da rede. Além disso, descobrimos que a atualização ideal da rede em várias etapas requer, em média, a mesma quantidade de controladores de uma migração em uma etapa, mas deve ter conhecimento total sobre cada etapa de transição da rede híbrida. Por fim, mostramos que o desempenho das abordagens heurísticas está relacionado à topologia e às características da rede, como tráfego e distribuição física dos *switches*.

**Palavras-chave:** problema de posicionamento do controlador, rede definida por software, planejamento de rede

# Abstract

SDN has emerged as an alternative networking paradigm separating the control plane from the data plane. In real-world scenarios, the transition from a traditional network to an SDN one is usually done in incremental steps, due to the costs and bureaucracy of new hardware's acquisition and deployment in companies. This transition involves the deployment of SDN-ready switches and hardware to run the software responsible for managing the control plane, i.e., the controller. Many studies on how to properly place and reduce the number of controllers, known as the *Controller Placement Problem* (CPP), have been conducted, but only a few of them exploited the CPP on hybrid (traditional/SDN) networks. In hybrid networks, we identified a gap in the minimum number of controllers required for a hybrid SDN to operate, considering a different number of transition steps and the heuristic to place these controllers within the network. Therefore, in this paper, we propose three heuristic approaches based on graph invariant for choosing the node to place a new controller, reducing its number on a network that starts legacy, becomes hybrid, and ends fully SDN-enabled at the end of a finite transition step horizon. The proposed heuristics are compared against an optimized model and have been proven useful in scenarios where the network administrator does not have full control over the network evolution. Also, we found that the optimal multi-step network upgrade requires, on average, the same amount of controllers of a one-step migration, but it has to have full knowledge on each transition change of the hybrid network. Finally, we show that the performance of the heuristic approaches is related to the network topology and characteristics, such as traffic and physical switches distribution.

**Keywords:** controller placement problem, software-defined networking, network planning

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Trabalhos Relacionados</b>	<b>3</b>
<b>3</b>	<b>Modelo do Sistema</b>	<b>5</b>
<b>4</b>	<b>Definição do Problema</b>	<b>7</b>
4.1	Restrições . . . . .	7
4.2	Problema de otimização do CPP . . . . .	9
<b>5</b>	<b>Proposta</b>	<b>11</b>
<b>6</b>	<b>Avaliação dos Resultados</b>	<b>14</b>
6.1	Análise dos resultados para carga otimizada . . . . .	17
6.2	Análise dos resultados para carga pesada . . . . .	18
<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>21</b>
	<b>Referências</b>	<b>23</b>
	<b>Apêndice</b>	<b>24</b>
<b>A</b>	<b>Artigo Publicado</b>	<b>25</b>



# Lista de Figuras

6.1	Topologias dos conjuntos de dados. . . . .	14
6.2	Carga [packet_in/s] como uma função do <i>throughput</i> da rede [pacotes/s]. . .	16
6.3	Número de controladores nos cenários de carga otimizada. . . . .	19
6.4	Número de controladores nos cenários de carga pesada. . . . .	19

# Lista de Tabelas

3.1	Notação usada nessa monografia. . . . .	6
6.1	Parâmetros para os cenários. . . . .	15

# Lista de Abreviaturas e Siglas

**BILP** *Binary Integer Linear Problem.*

**CPP** *Controller Placement Problem.*

**ISP** *Internet Service Provider.*

**MFC** Nó com Maior Fluxo de Carga.

**MLC** Mapeador de Localização do Controlador.

**NMC** Nó Mais Conectado.

**NME** Nó com Maior Excentricidade.

**NOC** *Network Operation Center.*

**QdE** Qualidade de Experiência.

**QdS** Qualidade de Serviço.

**SDN** *Software-Defined Network.*

**SNC** Seleccionador de Nós do Controlador.

# Capítulo 1

## Introdução

As Redes Definidas por Software (*Software-Defined Network* (SDN)) fornecem um provisionamento de rede logicamente centralizado, com capacidade de programação e baixo custo de operações, surgindo como uma alternativa para atender aos rigorosos requisitos de rede atuais [1] [2]. SDNs requerem um novo tipo de componente de software geralmente implantado em hosts de processamento dedicados: os controladores. Estes controladores são responsáveis por gerenciar os fluxos da rede e prover o estado da rede trazendo benefícios como roteamento otimizado e melhor desempenho da rede [3], contribuindo para atender a Qualidade de Experiência (QdE) dos clientes.

A implantação de SDN em uma organização é um processo que não acontece de uma vez. Em vez disso, a rede legada evolui gradualmente para uma rede totalmente SDN [4]. Portanto, existem várias versões intermediárias de uma rede legada/SDN, chamadas de híbridas. Nessa evolução, cada nova versão da rede evolui da versão anterior influenciada por diversos fatores, como orçamento, recursos de equipamento, políticas de rede, estratégias de gerenciamento e domínios administrativos. Nesta monografia, estamos interessados em observar a adoção gradual de SDN em redes operadas por equipes de gestão distintas e independentes. Em tal rede, cada equipe pode tomar decisões de implantação de SDN independentemente umas das outras. Por exemplo, isso acontece em algumas universidades, onde cada departamento ou instituto tem flexibilidade para fazer melhorias locais em sua rede interna com orçamento próprio. No mundo corporativo, isso pode ser observado em uma empresa onde cada filial controla quando atualizar sua sub-rede que se conecta à matriz.

O controlador SDN [5] é um componente crítico de uma rede SDN. Naturalmente, as decisões do projeto SDN em relação a este componente têm um impacto direto no desempenho, carga e custos da rede [6]. Considerando o projeto de uma SDN, destacamos três decisões fundamentais sobre os controladores: (*i*) a quantidade mínima de controladores instalados para reduzir custos sem prejudicar o desempenho da rede; (*ii*) a localização

dos controladores, levando em consideração a demanda da rede; e (iii) quando um novo controlador deve ser implantado, seguindo a linha do tempo de evolução da rede [7]. Implantar uma SDN considerando essas decisões de projeto não é simples, principalmente quando se deve considerá-las todas juntas, levando ao chamado Problema de Posicionamento do Controlador (*Controller Placement Problem* (CPP)). As decisões (i) e (ii) foram amplamente abordadas na literatura. A decisão (iii), no entanto, é pouco explorada, mas argumentamos que uma organização não pode implantar SDN adequadamente ignorando uma abordagem evolucionária. Seguir uma implantação instantânea, como geralmente assumido na literatura, não é realista.

Existem muitos estudos considerando o problema de posicionamento do controlador [8]. No entanto, a maioria se concentra em lidar com o CPP como um problema de otimização de objetivo único, trabalhando com recursos como custo, resiliência e latência [9]. Outros tentam modelar redes híbridas usando hardware que ainda não foi produzido [10]. No entanto, as decisões sobre a quantidade mínima, a melhor localização e o momento para colocar um controlador foram analisadas separadamente, sem considerar sua relação. Até onde sabemos, esta é a primeira análise do CPP que aborda a relação entre essas decisões fundamentais mencionadas em uma rede híbrida.

Nesta monografia, abordamos o problema de posicionamento do controlador, considerando o número mínimo de controladores necessários para fazer a transição de uma rede tradicional para uma SDN híbrida até que se torne uma rede totalmente SDN. Nossa solução toma decisões com relação ao tempo, localização e quantidade mínima de controladores SDN necessários para atender às restrições de latência e carga. As principais contribuições desta monografia são listadas a seguir:

- Uma caracterização do problema de posicionamento do controlador em relação ao tempo, número e localização dos controladores em SDNs híbridas;
- Um modelo analítico para avaliar os limites de *trade-off* relacionados ao número, localização e tempo para implantar controladores em uma SDN híbrida; e
- Uma análise experimental do *trade-off* de tempo, número e localização dos controladores em SDNs híbridas.

O restante desta monografia está organizado da seguinte forma. No Capítulo 2, resumimos e discutimos trabalhos relacionados. No Capítulo 3, apresentamos nosso modelo de sistema. No Capítulo 4, definimos nosso modelo de otimização. No Capítulo 5, um modelo experimental é apresentado. Posteriormente, no Capítulo 6, apresentamos e discutimos os resultados de nossos experimentos. No Capítulo 7, concluímos nosso trabalho apresentando conclusões e trabalhos futuros. Finalmente, no Apêndice A, são apresentados os trabalhos científicos resultantes deste trabalho.

# Capítulo 2

## Trabalhos Relacionados

Este capítulo apresenta os trabalhos relacionados mais relevantes para resolver o problema de posicionamento do controlador.

Amir Hossein Fakhteh *et al.* [11] trabalham com foco em um algoritmo para a migração de uma rede tradicional para uma rede definida por software, com foco na maximização da capacidade de controle e flexibilidade da rede, minimizando o custo dessa transição. Os autores usam a *closeness-centrality* e *neighbor eccentricity*, que são invariantes da teoria dos grafos, para selecionar heurísticamente os melhores nós a serem atualizados e, em seguida, posicionar o controlador com base na posição do *cluster* ponderada pelo número de *switches* atualizados nos *clusters*. Como nosso trabalho, [11] explora o CPP em SDN híbrida, mas eles lidam apenas com a localização de um controlador, e assumem que a transição da rede é previsível e controlada pelo gerente da rede. Diferentemente, nosso trabalho lida com múltiplos controladores em redes onde o gerente não tem controle centralizado sobre a evolução dos *switches*, criando cenários aleatórios para avaliar o caso médio. Uma vez que a principal característica de uma SDN é a capacidade de reprogramar todo o plano de controle de acordo com a própria vontade do gerente a qualquer momento, pretendemos minimizar o número de controladores que produzem melhorias de longo prazo otimizando o planejamento da rede em vez de seu desempenho.

Levente Csikor *et al.* [12] virtualizam os *switches* SDN, desenvolvendo um novo hardware acoplado a *switches* legados para torná-los SDN. Essa solução provou custar menos e ter melhor desempenho no plano de dados para pequenas empresas em comparação com a implantação de *switches* SDN existentes, mas pode não ser uma solução definitiva a longo prazo. Ele fornece uma maneira de atualizar uma rede legada para SDN. Nosso trabalho visa uma SDN completa como o resultado final, com maior polarização na topologia da rede. Além disso, nossos resultados são generalizados para atingir o caso médio e não específico para um caso de uso único.

Damu Ding *et al.* [13] propõem uma solução para compor e monitorar uma rede

com *switches* legados e SDN. Os autores pretendem implantar *switches* SDN de forma incremental em uma rede ISP para maximizar o número de fluxos distintos monitorados. Para atingir este objetivo, os autores propuseram um algoritmo e uma nova detecção de *heavy-hitter* em toda a rede que funcionou bem em redes com poucos *switches* legados. Tanto a detecção de *heavy-hitter* quanto o algoritmo de implantação incremental superaram as abordagens existentes, considerando o plano hierárquico de atualização da rede. Da mesma forma, definimos um horizonte finito de etapas discretas incrementais para modelar nossa transição de uma rede legada para uma rede totalmente SDN. No entanto, damos um passo adiante caracterizando a compensação de posicionamento, número e tempo para implantar controladores SDN para o caso médio, randomizando os *switches* legados que se tornarão habilitados para SDN.

Tamal Das e Mohan Gurusamy [7, 9] têm dois trabalhos que introduzem o CPP para adicionar controladores de forma incremental em uma rede ao longo de um período de tempo. Em [7], eles avaliam sua solução em termos de várias métricas de desempenho, como o número de controladores, economia de custos e latência. O modelo proposto se concentra na otimização dos custos de posicionamento e latência do *switch* para o controlador, considerando o aumento do tráfego de rede e a queda dos custos de hardware e manutenção SDN. Isso é feito primeiro minimizando o custo de posicionamento do controlador, assumindo um limite de latência e, em seguida, com a saída do primeiro estágio, eles reduzem a latência de controle do pior caso. Em [9], eles formulam o CPP em SDN híbrida, com o objetivo de maximizar a resiliência do canal de controle. Ao contrário deste trabalho, consideramos a otimização da latência do *switch* para o controlador um objetivo incomum, visto que o desempenho da rede está relacionado à demanda e não há garantias de que minimizando o tempo de ida e volta da sinalização entre o controlador e os *switches* trará benefícios para a operação da rede. Assim, avaliamos o CPP para o caso médio de uma dada topologia modelando a latência do *switch* para o controlador como uma restrição.

Considerando a literatura apresentada, a maior parte dos trabalhos se concentram na otimização de um objetivo, como latência ou resiliência, mas não discutem as compensações envolvendo tempo, número e localização de controladores, criando uma lacuna de pesquisa: quantificar o *trade-off* envolvendo tempo, número e localização de controladores na transição de uma rede legada para SDN, levando em consideração a latência do *switch* para o controlador e as restrições de carga do controlador. Esta monografia cobre essa lacuna, considerando o caso médio em que a evolução da rede não é previsível. Esse resultado é então usado como linha de base para comparar três heurísticas que propomos para a escolha do controlador localizado no mesmo cenário SDN híbrido, conforme apresentado nos próximos capítulos.

# Capítulo 3

## Modelo do Sistema

Este capítulo apresenta o modelo do sistema e outras definições para o problema de posicionamento do controlador usado neste trabalho. A Tabela 3.1 reúne todas as notações usadas neste capítulo. Nosso modelo começa com a definição da rede legada, que é representada pelo conjunto  $\mathcal{N} = \{1, 2, \dots, N\}$  nós, por exemplo, *switches*. Cada par de nós  $(n, m)$  constitui uma borda conectada quando  $\{(n, m) \in \mathcal{N} \mid n \neq m; \varepsilon(n, m) = 1\}$ , onde  $\varepsilon(n, m)$  representa a função de borda e assume o valor de 1 apenas quando ambos os nós  $n$  e  $m$  possuem uma conexão física entre eles e 0 caso contrário. A função de borda pode ser iterada considerando a permutação de todos os nós  $\mathcal{N}$  para gerar a matriz binária das arestas definidas  $V = \{0, 1\}_{N \times N}$ . Finalmente, uma topologia de rede pode ser definida pelo grafo direcionado  $G(\mathcal{N}, V)$ .

A atualização de uma rede legada  $G(\mathcal{N}, V)$  para SDN é realizada em um horizonte finito de  $\mathcal{T} = \{1, 2, \dots, T\}$  transições discretas  $t \rightarrow \mathbb{N}$ . Cada etapa de transição  $t \in \mathcal{T}$  é determinada como uma atualização de pelo menos um ou mais nós legados para SDN. Uma matriz binária  $X = \{\forall t \in \mathcal{T}, \forall n \in \mathcal{N} \mid X_{tn} \in \{0, 1\}\}$  é usada para rastrear quais nós estão se tornando SDN, onde  $X_{tn}$  só é igual a 1 quando o nó  $n$  na etapa de transição  $t$  está habilitado para SDN. Depois que um nó  $n$  torna-se habilitado para SDN, ele nunca mais voltará a ser um equipamento legado. Por isso a propriedade  $X_{(t+1)n} = \{1 \mid X_{tn} = 1\}$  é válida para qualquer configuração de  $X$ .

Os nós legados e habilitados para SDN recebem tráfego como uma carga de rede, representada pelo vetor  $K = \{\forall n \in \mathcal{N} \mid K_n \in \mathbb{Z}^+\}$ . Cada elemento  $K_n$  é um valor inteiro positivo do número absoluto de pacotes recebidos no nó  $n$ . Como o controle do equipamento legado é realizado localmente, apenas os nós habilitados para SDN geram cargas de trabalho de sinalização. Embora a carga de trabalho de sinalização deva ser apresentada em *packets\_in*, os conjuntos de dados disponíveis na literatura fornecem apenas valores de carga de trabalho de rede. Para resolver este problema, estamos usando a representação da carga de trabalho de sinalização como uma função direta dos pacotes



Tabela 3.1: Notação usada nessa monografia.

Notação	Parâmetro
$G(\mathcal{N}, V)$	Grafo da rede
$\sigma$	Carga máxima de um controlador
$K_n$	Carga (número de pacotes) de um nó $n$
$L$	Matriz de latências do conjunto de nós $\mathcal{N}$
$L_{nm}$	Caminho de latência mais curta entre os nós $n$ e $m$
$\delta$	Limite máximo de latência entre um <i>switch</i> e seu controlador
$\mathcal{N}$	Conjunto de nós
$\mathcal{T}$	Horizonte finito de etapas de transição
$T$	Cardinal do conjunto de etapas de transição
$V$	Conjunto de arestas
$X$	Matriz binária
$X_{tn}$	Define se um nó $n$ é SDN na etapa de transição $t$
$\varepsilon(n, m)$	Representação de conexão entre os nós $n$ e $m$

recebidos por nó como  $f(K_n) \in \mathbb{Z}^+$  *packet\_in*. Um controlador implantado irá processar uma carga de trabalho máxima de  $\sigma \in \mathbb{Z}^+$  *packet\_in* somada de todos os nós SDN atribuídos a ele.

A rede definida em  $G(\mathcal{N}, V)$  tem sua latência denotada pela matriz  $L = \{\forall(n, m) \in \mathcal{N} | L_{nm} \in \mathbb{R}^+\}$ , onde cada elemento  $L_{nm}$  é igual a um número real positivo que representa a latência somada resultante da solução do problema do caminho mais curto entre os nós  $n$  e  $m$ , caso contrário, 0 quando  $n = m$ . Além disso, toda latência de comunicação de sinalização de controle SDN deve ser mantida abaixo de um limite de  $\delta$  para atender aos requisitos de Qualidade de Serviço (QoS), conhecidos como latência máxima de propagação do *switch* para o controlador [8].

O modelo de sistema descrito foi projetado para representar a maioria dos cenários reais de rede com topologias gerais, considerando os principais indicadores de desempenho, como tráfego e latência. Além disso, um horizonte finito compreendendo etapas de transição também foi introduzido para mapear a atualização de uma rede legada em uma rede totalmente habilitada para SDN. O modelo proposto também pode representar a maioria dos conjuntos de dados de rede da literatura, como os considerados em nosso trabalho, NSFNet, GEANT2 e SYNTH50 [14]. A seguir, considerando o modelo de sistema proposto, definimos o problema do CPP.

# Capítulo 4

## Definição do Problema

Neste capítulo, descrevemos a definição do problema CPP. Começamos definindo as principais variáveis de decisão, seguidas das restrições e objetivos do CPP. Finalmente, uma definição de Problema Linear Inteiro Binário (*Binary Integer Linear Problem* (BILP)) é proposta discutindo diferenças em relação aos modelos encontrados na literatura.

O principal objetivo do CPP é minimizar o número de controladores. O aspecto diferente do nosso modelo é reduzir o número de controladores considerando a transição de uma rede legada até que ela seja totalmente habilitada para SDN. Como consequência, as decisões sobre a colocação do controlador são descritas pelas variáveis binárias  $y_{tn}$ , onde

$$y_{tn} = \begin{cases} 1; & \text{Controlador no nó } n \text{ está instalado em } t; \\ 0; & \text{Caso contrário.} \end{cases}$$

Da mesma forma, cada decisão de atribuição de um *switch* SDN para um controlador pode ser mapeada usando variáveis binárias  $z_{tnm}$ , onde

$$z_{tnm} = \begin{cases} 1; & \text{Nó } n \text{ associado ao controlador } m \text{ em } t; \\ 0; & \text{Caso contrário.} \end{cases}$$

Considerando cada variável binária na etapa  $t$ , podemos decidir quando um nó  $n$  começa a hospedar um controlador SDN com  $y_{tn}$  e quais nós  $m$  ele está servindo com  $z_{tnm}$ . Cada decisão deve ser tomada considerando diferentes restrições, conforme apresentado a seguir.

### 4.1 Restrições

- O controlador precisa ser executado em um novo hardware ou virtualizado no hardware existente. Portanto, é mais conveniente, devido à infraestrutura existente,

colocar um controlador onde já temos um *switch*. Assim, esta restrição permite que um controlador seja colocado ( $y_{tn} = 1$ ) apenas se o nó  $n$  na etapa  $t$  for SDN ( $X_{tn} = 1$ ).

$$y_{tn} \leq X_{tn}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}. \quad (4.1)$$

- Os *switches* SDN enviam mensagens *packet\_in* aos controladores sempre que chega um pacote com cabeçalho desconhecido, ou seja, com campos que não se enquadram em nenhuma regra de encaminhamento já instalada. Portanto, a latência ( $L_{nm}$ ) em segundos entre o controlador no nó  $m$  e um nó SDN  $n$ , quando atribuído ( $z_{tnm} = 1$ ) deve estar abaixo de um limite de atraso de  $\delta$  segundos para manter os requisitos de QoS.

$$z_{tnm} \times L_{nm} \leq \delta, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}, \forall m \in \mathcal{N}. \quad (4.2)$$

- Cada *switch* SDN precisa ser atribuído a um controlador para operar, recebendo regras de encaminhamento. Para a etapa de transição  $t^{th}$ , um nó SDN ( $X_{tn} = 1$ )  $n$  deve ser associado a um e apenas um controlador  $m$  ( $\sum_{m=1}^{\mathcal{N}} z_{tnm} = 1$ ).

$$\sum_{m=1}^{\mathcal{N}} z_{tnm} = X_{tn}, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}. \quad (4.3)$$

- Como a latência entre o nó SDN associado e o controlador será insignificante, é natural garantir que o *switch* hospedeiro esteja conectado ao controlador hospedado. Portanto, o nó  $n$  que hospeda um controlador ( $y_{tn} = 1$ ) na etapa  $t$  sempre será atribuído ao seu controlador hospedado ( $z_{tnn} = 1$ ).

$$z_{tnn} = y_{tn}, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}. \quad (4.4)$$

- Um nó SDN deve ser atribuído a um controlador que deve existir em outro nó. Assim, essa restrição garante que um *switch* SDN  $n$  só será atribuído ( $z_{tnm} = 1$ ) a um nó com um controlador ( $y_{tm} = 1$ ) na etapa  $t$ .

$$y_{tm} - z_{tnm} \geq 0, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}, \forall m \in \mathcal{N}. \quad (4.5)$$

- Devido às limitações de hardware, os controladores só podem lidar com uma determinada carga de trabalho de *packet\_in* por segundo. Esta restrição garante que o somatório de *packet\_in* recebido ( $K_n$ ) de todos os nós atribuídos ( $\sum_{n=1}^{\mathcal{N}} z_{tnm}$ ) deve

ser menor que o limite da carga de trabalho do controlador  $\sigma$ .

$$\sum_{n=1}^N z_{tnm} \times K_n \leq \sigma, \forall t \in \mathcal{T}, \forall n \in \mathcal{N}. \quad (4.6)$$

- Comprar, colocar ou substituir um controlador gera custo monetário. No nosso modelo, uma vez que um controlador é colocado ( $y_{tn} = 1$ ), ele não pode ser removido ou substituído e permanece no mesmo local até o final de todas as etapas de transição futuras ( $y_{(t+1)n} = 1$ ).

$$-1 \leq y_{tn} - y_{(t+1)n} \leq 0, \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \mid t < T. \quad (4.7)$$

A restrição (4.2) lida com o aspecto de localização em nosso modelo. Considerando as variáveis de decisão e restrições, a função objetivo do CPP pode ser definida como uma minimização do número de controladores, conforme definido pela Equação (4.8).

## 4.2 Problema de otimização do CPP

$$\min \sum_{t=1}^T \sum_{n=1}^N y_{tn} \quad (4.8)$$

s.a.

$$(4.1), (4.2), (4.3), (4.4), (4.5), (4.6), \text{ e } (4.7).$$

Na Equação (4.8), a instalação de um novo controlador é caracterizada pela primeira ocorrência da variável  $y_{tn}$ , ou seja, para um determinado nó  $n$ , é o menor valor que  $t$  pode assumir, onde  $y_{tn} = 1$ . É assim que lidamos com a parte do tempo. Devido ao efeito de propagação causado pela restrição (4.7) que torna todos  $y_{(t+1)n} = 1$  até  $t = T - 1$  de sua primeira ocorrência, a minimização pode ser generalizada para a ocorrência de todas as variáveis  $y_{tn}$  em todas as etapas de transição. Assim, nosso objetivo é a minimização generalizada da ocorrência de controladores para cada nó SDN em  $G(\mathcal{N}, V)$ . Como o modelo proposto é um Problema Linear Inteiro Binário (BILP), ele se reduz a um problema combinatório de complexidade não polinomial. Embora combinatório, nosso problema apresenta um objetivo linear e restrições, e a solução pode ser encontrada por solucionadores de programação linear, como C-PLEX<sup>1</sup> ou OR-Tools<sup>2</sup>.

Para resolver o problema de otimização proposto, ele deve conhecer todas as informações disponíveis antes de iniciar. Isso significa que o número de etapas de transição é finito e conhecido. Também se sabe quais nós serão atualizados para SDN em cada etapa.

<sup>1</sup>C-PLEX-IBM Inc.- <https://www.ibm.com/br-pt/analytics/cplex-optimizer>

<sup>2</sup>OR-Tools - Google Inc. - <https://developers.google.com/optimization>

No entanto, em implantações não planejadas, não controladas ou em cenários reais, essas afirmações não podem ser observadas. Por exemplo, em um ambiente universitário, diferentes departamentos podem escolher o número e os nós legados que serão atualizados sem notificação prévia ao *Network Operation Center* (NOC). Assim, os gerentes de NOC podem não ter todas as informações necessárias para resolver esse problema de otimização. Assim, eles não poderão planejar como distribuir e instalar de forma otimizada novos controladores na infraestrutura da Universidade. Consequentemente, os gerentes de rede terão que assumir uma estratégia abaixo do ideal ou alguma política para implantar novos controladores para toda a Universidade. Eles seguirão esta abordagem a fim de manter algum nível de controle dos fluxos da rede acadêmica, apesar das diferentes decisões de atualização de cada departamento.

# Capítulo 5

## Proposta

Resolver o problema de otimização proposto torna viável identificar o número mínimo, a localização e em que etapa de transição os controladores devem ser instalados para uma dada topologia. No entanto, essa abordagem torna-se inviável para cenários não controlados, onde as etapas das transições ocorrem sem planejamento prévio. Neste caso, propomos três soluções heurísticas para escolher onde colocar um novo controlador, cada uma baseada em uma política diferente considerando uma invariante de grafo. Essas políticas de posicionamento do controlador são:

- Nó Mais Conectado (NMC);
- Nó com Maior Excentricidade (NME); e
- Nó com Maior Fluxo de Carga (MFC).

Na política NMC, o nó com o maior grau disponível é selecionado. Esta seleção visa diminuir o número médio de saltos entre os controladores e nós colocados, considerando a intuição de [15]. Ao contrário da política anterior, NME seleciona o nó com a maior excentricidade, ou seja, esta política maximiza a distância do nó escolhido a todos os outros na topologia. Por último, a política MFC seleciona um nó com base em sua carga (*packet\_in/s*). Como a carga é uma função direta do tráfego do *switch* experimentado, colocar um novo controlador no nó de carga mais alta diminuirá o tráfego de sinalização e diminuirá a latência do controlador para o *switch*.

Antes de implementar cada política, propomos o Algoritmo 1 que apresenta o algoritmo Mapeador de Localização do Controlador (MLC). Este algoritmo é responsável por identificar se um novo controlador é necessário ou não. Essa decisão é baseada na chegada de novos nós SDN, na carga da rede e nas restrições de latência do *switch* para o controlador. MLC também é responsável por chamar o algoritmo Seleccionador de Nós do Controlador (SNC) (linha 3) apresentado no Algoritmo 2. O SNC selecionará um nó SDN para instalar um novo controlador, com base em uma das três políticas disponíveis.

---

**Algoritmo 1:** Mapeador de Localização do Controlador (MLC)

---

**entrada:**  $\mathcal{N}$ ,  $t$  e *política*  
**saída** : Uma lista  $l$  do mapeamento dos controladores para o  $t$  atual

```
1 mapear_switches( $t$ );
2 enquanto algum switch SDN não está mapeado e contador_limite <  $\mathcal{N}$  faça
3   |   node = selecionador_de_nós_do_controlador( $\mathcal{N}$ ,  $t$ , política);
4   |   inserir_controlador(node,  $t$ );
5   |   mapear_switches( $t$ );
6 fim
7 se algum nó SDN não está mapeado então
8   |   parar;
9 fim
10 retorna  $l$ ;
```

---

O algoritmo MLC recebe como entrada o conjunto de nós  $\mathcal{N}$ , a etapa de transição atual  $t$  e a *política* em uso. O valor de *política* pode ser uma das heurísticas de seleção definidas como NMC, NME e MFC. Como definimos que uma nova etapa  $t$  significa uma atualização de um ou mais *switches*, toda execução do Algoritmo 1 começa chamando a função *mapear\_switches()* (linha 1). Esta função reconfigura o mapeamento de associação do controlador de cada *switch* para a etapa  $t$  atual. Uma vez que a carga em um controlador depende apenas dos *switches* mapeados para ele, o processo de re-mapeamento dos *switches* para os controladores localizados em locais diferentes não afeta a carga do controlador, apenas a latência do *switch* para o controlador. No caso de um ou mais *switches* SDN permanecerem sem serem atribuídos, devido à latência ou restrições de carga, o algoritmo entra no loop *while* (linha 2). Assim, o MLC selecionará um nó, com base na *política* (linha 3) em execução no momento, para colocar o controlador (linha 4), até que todos os *switches* SDN sejam atribuídos. O processo de mapeamento é atualizado considerando um novo controlador colocado (linha 5) ou novos *switches* SDN (linha 1). No final da execução, o MLC retorna o lugar de todos os controladores e o mapeamento do controlador de cada *switch* para a etapa atual (linha 10). Se houverem *switches* SDN não atribuídos a nenhum controlador no final da execução, o problema é inviável.

O Algoritmo 2 apresenta a implementação do SNC. Ele recebe como parâmetro de entrada o conjunto de nós  $\mathcal{N}$ , as etapas de transição atuais  $t$  e a *política* usada. Com base na *política* e no passo de transição atual  $t$  (linha 2), o SNC itera sobre todos os nós SDN (linha 1), considerando para isso os nós SDN sem um controlador instalado nele (linha 2). Para cada nó, o SNC verificará se a posição do nó maximiza o valor da métrica da *política* (linha 3). No final da execução, o algoritmo retornará (linha 9) o nó SDN que maximiza o valor avaliado da métrica da *política* (linhas 3 a 5).

---

**Algoritmo 2:** Seleccionador de Nós do Controlador (SNC)

---

**entrada:**  $\mathcal{N}$ ,  $t$  e *política*  
**saída** : Nó escolhido

```
1 para  $n \leftarrow 1$  até  $\mathcal{N}$  faça
2   se ( $n$  é SDN em  $t$ ) e ( $n$  não hospeda um controlador) então
3     se  $métrica(n, política) > maior\_valor$  então
4        $no\_escolhido = n$ ;
5        $maior\_valor = métrica(n, política)$ ;
6     fim
7   fim
8 fim
9 retorna  $no\_escolhido$ ;
```

---

Todos os algoritmos propostos foram implementados usando a linguagem de programação Python. Esta linguagem nos permitiu utilizar uma biblioteca específica e otimizada para análise de grafos, chamada NetworkX<sup>1</sup>. O NetworkX foi usado para construir a topologia do grafo da rede e para calcular as operações do grafo, como grau de nó e excentricidade. Além disso, o problema de otimização foi resolvido usando OR-Tools da Google Inc. A seguir, apresentamos os resultados obtidos considerando diferentes conjuntos de dados.

---

<sup>1</sup>NetworkX <https://networkx.github.io>



# Capítulo 6

## Avaliação dos Resultados

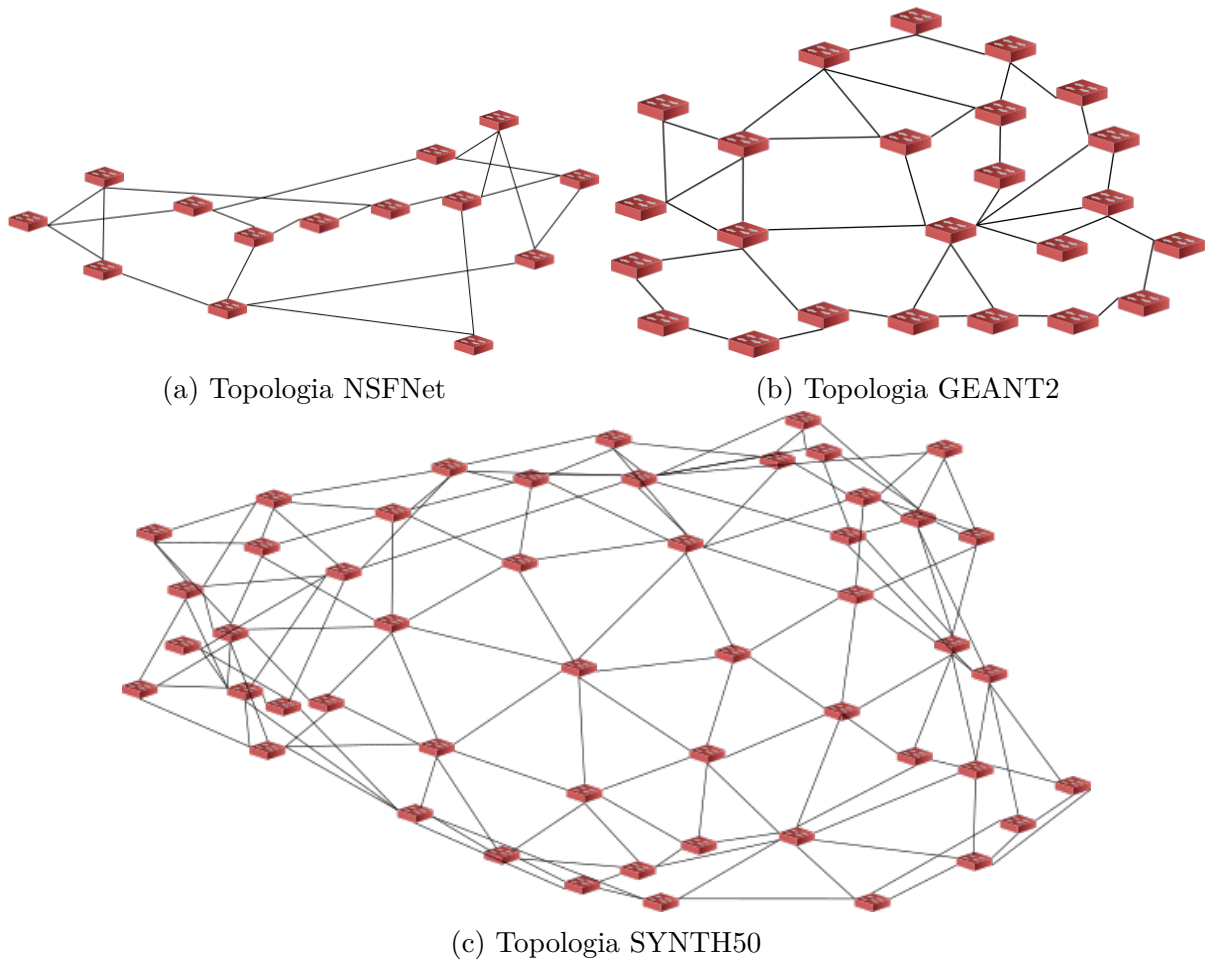


Figura 6.1: Topologias dos conjuntos de dados.

Para avaliar o problema de otimização em comparação com as heurísticas selecionadas, usamos três conjuntos de dados [14]: (i) NSFNet (14 nós), (ii) GEANT2 (24 nós) e (iii)

Tabela 6.1: Parâmetros para os cenários.

Parâmetro	NSFNet	GEANT2	SYNTH50
$\mathcal{N}$	14	24	50
$\mathcal{T}$	{1, 5, 10}	{1, 5, 10, 15, 20}	{1, 5, 10, 15, 20}
$L_{nm}$	[0, 1.55684]	[0, 0.734737]	[0, 0.734347]
$\delta$	0.25	0.25	0.25
$K_n$	[0, 14340]	[0, 9221]	[0, 16217]
$\sigma$	20000	20000	20000
$X_{tn}$	Aleatório	Aleatório	Aleatório

SYNTH50 (50 nós). As topologias dos conjuntos de dados selecionados são ilustradas na Figura 6.1. Conduzimos experimentos considerando a otimização e as três heurísticas propostas com NSFNet e GEANT2. Posteriormente, para testar cada heurística em um ambiente mais desafiador, selecionamos um conjunto de dados de  $N = 50$  nós, o SYNTH50 [14]. Porém, como o SYNTH50 apresenta um grande número de nós e consequentemente um número ainda maior de variáveis a serem consideradas, o tempo de execução do otimizador torna-se impraticável e seus resultados não são mostrados.

O conjunto de dados NSFNet foi pesquisado e detalhado, descrito em [16]. Uma vez que todos os conjuntos de dados considerados seguem a mesma metodologia, a latência média experimentada entre os nós é em média 40ms, precisamente em [16, Figura 7]. Nesse sentido, adotamos o valor da latência máxima do *switch* para o controlador para o valor normalizado de 0,25, que corresponde a aproximadamente 10ms nesses cenários. Além disso, adotamos 20000 *packet\_in/s* para a capacidade máxima do controlador de acordo com [17, 18, 19, 20]. Mais detalhes são apresentados na Tabela 6.1.

Como pode ser visto na Tabela 6.1, cada um dos conjuntos de dados apresenta uma topologia e configuração diferente.  $\mathcal{N}$  representa o número de nós.  $\mathcal{T}$  é o conjunto de etapas de transição.  $\delta$  e  $\sigma$  são a latência máxima do controlador para o *switch* e as restrições de carga de trabalho máxima do controlador, respectivamente.  $L_{nm}$  e  $K_n$  são o intervalo dos valores para a latência normalizada e o número de pacotes por segundo no conjunto de dados.  $X_{tn}$  é a transição do *switch* SDN gerada aleatoriamente para cada etapa.

Ao longo de nossos experimentos, dividimos a carga da rede em duas categorias: carga otimizada e carga pesada. Para a carga otimizada, contamos com os resultados da estratégia de redução de *overhead* desenvolvida no trabalho [21], que caracteriza o número de *packet\_in* em termos de número de pacotes. Realizamos uma interpolação linear no gráfico em [21, Figure 8] resultando em uma função que usamos para converter o número de pacotes nos conjuntos de dados em *packet\_in* apresentadas na Figura 6.2, a que nos referimos como carga. A carga pesada é onde um pacote na rede corresponde a um

$packet\_in$  no controlador. A carga em um *switch* é a soma do número absoluto de pacotes transmitidos de todos os *switches* adjacentes para este *switch*.

Para uma generalização das mudanças de rede, o plano de atualização é criado aleatoriamente com um ou mais *switches* legados se tornando SDN a cada etapa de transição até que todos se tornem habilitados para SDN. Executamos 35 rodadas de cada experimento para atingir um intervalo de confiança de 95%. Para avaliar o impacto do tempo, cada experimento é avaliado para diferentes horizontes de etapas de transição que abrangem o conjunto  $\mathcal{T}' = \{1, 5, 10, 15, 20\}$ . Dado que uma etapa significa pelo menos uma atualização de um nó legado para SDN, paramos de experimentar com o valor mais alto que  $t' \in \mathcal{T}'$  pode assumir que é menor do que o número total de nós por conjunto de dados. Por exemplo, NSFNet é um conjunto de dados de uma rede de 14 nós, então executamos o experimento até  $t' = 10$ .

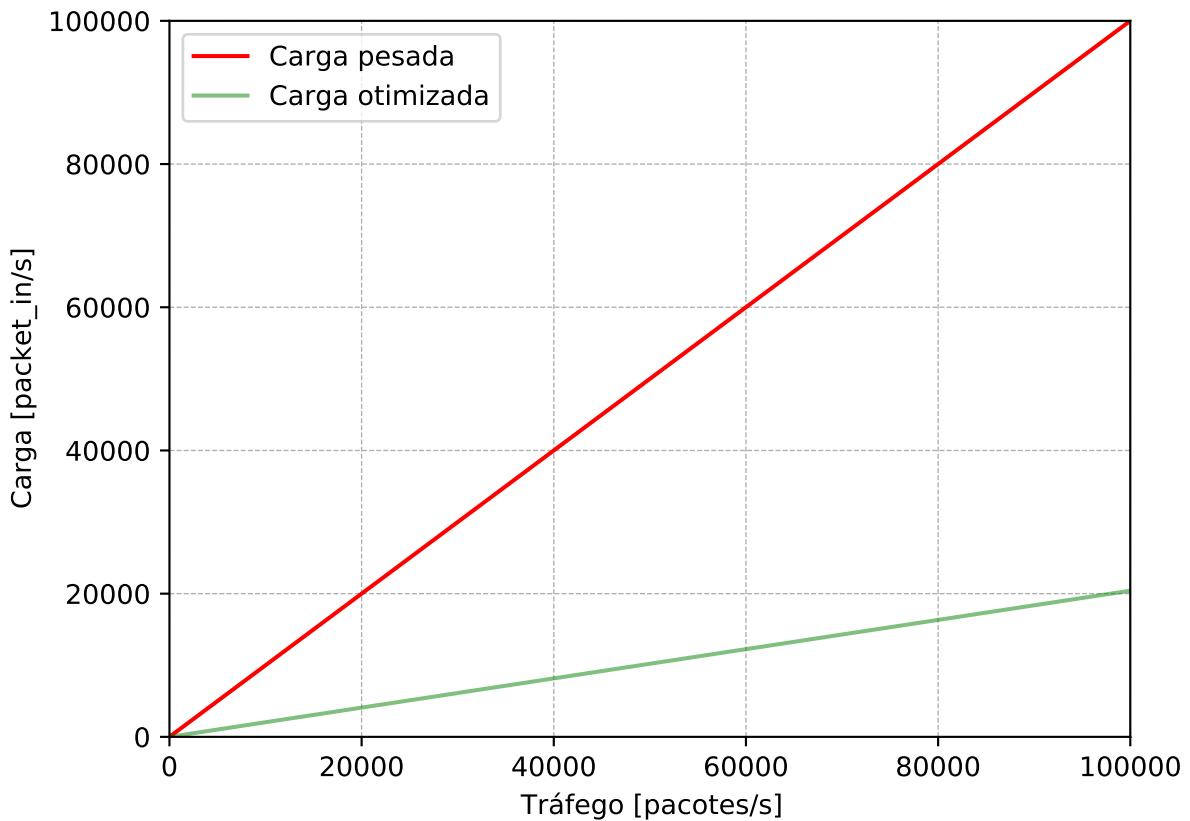


Figura 6.2: Carga [packet\_in/s] como uma função do *throughput* da rede [pacotes/s].

Apresentaremos primeiro os resultados dos experimentos para os conjuntos de dados. Eles são divididos em cenário otimizado e de carga pesada. Como a carga média do controlador é complementar ao número mínimo de controladores, decidimos omiti-los.

## 6.1 Análise dos resultados para carga otimizada

A principal métrica apresentada ao longo de nossos experimentos é o número médio de controladores no final do horizonte finito de etapas de transição. Na Figura 6.3, temos no eixo vertical o número médio de controladores sendo apresentado. No eixo horizontal, temos o valor cardinal de diferentes conjuntos de etapas de transição  $T$  sendo apresentados. Cada valor de  $T$  contém um grupo de barras, uma para cada heurística e a solução otimizada. Como  $T = 1$  significa uma transição de legado para SDN em uma única etapa, as variáveis  $X_{tn}$  são sempre 1 para cada *switch*. Nesse caso, os resultados das abordagens otimizada e heurística não apresentam variações, pois não há inserção de incerteza causada pelos passos aleatórios presentes nos demais casos. Portanto, não há barras de erro nas primeiras colunas de cada resultado apresentado. Além disso, dado que as heurísticas são sub-ótimas, sem reajustes, elas podem convergir para soluções sub-ótimas que não necessariamente apresentam o melhor resultado.

Como esperado, o modelo otimizado proposto sempre requer a menor quantidade de controladores e tem uma maior utilização do controlador, atingindo um valor médio conciso de 4 controladores para o NSFNet, exibida na Figura 6.3a. Comparada à otimização, a melhor heurística para este caso de uso é o MFC, atingindo o valor médio de 5,56 controladores, cerca de dois controladores a mais que a otimização. Por outro lado, a pior heurística a ser considerada é a NME, com 10 controladores necessários para o pior caso e cerca de 6,20 no melhor caso, cerca de 3 controladores a mais que a otimização. A política NMC apresenta um comportamento semelhante à heurística NME.

Dado que o NSFNet é uma pequena rede de 14 nós não totalmente conectada, colocar o controlador no nó mais carregado em MFC traz benefícios em relação aos controladores que processam a carga de trabalho localmente. Percebe-se ainda que usar a maior excentricidade do NME não explora o mesmo benefício e aumenta a chance de violação da restrição de latência, como apresentado no intervalo de  $L_{nm}$  na Tabela 6.1. Outro ponto importante é que com o aumento do número de etapas de transição, as três heurísticas passam a apresentar resultados semelhantes, possibilitando a utilização de qualquer uma das políticas consideradas para uma rede com a mesma topologia e um grande tempo de implantação.

Na Figura 6.3b, o número mínimo de controladores para GEANT2 é representado pela otimização, apresentando uma média de 3,4 controladores no pior caso. A melhor política para a rede GEANT2 é a heurística NME, apresentando sete controladores no pior caso e 5,4 na média no melhor. Por outro lado, a pior política é a MFC apresentando na pior das hipóteses 15 controladores e atingindo o mesmo valor de 9 controladores que o NMC para 20 etapas de transição.

Como GEANT2 apresenta uma topologia altamente conectada, posicionar os controladores nos nós de maior excentricidade, como em NME, permite que a atribuição entre controladores e *switches* SDN seja melhor executada, reduzindo o número de saltos entre eles. No entanto, uma heurística baseada na carga mais alta, como no MFC, apresenta desempenho inferior em GEANT2 devido à sua topologia altamente conectada, reduzindo a distância entre os controladores, aumentando o número de saltos por atribuição e aumentando a chance de violar as restrições de latência. Além disso, significa que a carga de um nó individual é semelhante aos demais em GEANT2, o que é o oposto do que acontece com o NSFNet, como observado na Tabela 6.1 em  $K_n$ . Para confirmar essa afirmação, executamos experimentos com o SYNTH50. Além disso, vale ressaltar que o aumento das etapas de transição reduz a diferença de uma heurística para outra, mas diferentemente do NSFNet, a política NME para GEANT2 continua melhor apesar do maior tempo de implantação.

SYNTH50 é uma topologia altamente conectada com 50 nós. Conforme mencionado, o número de variáveis aumenta drasticamente, tornando o tempo de execução da otimização inviável. Assim, apenas as políticas NMC, NME e MFC apresentam resultados considerando este conjunto de dados. Como pode ser visto, a política NME apresenta os melhores valores entre as três heurísticas com 9 controladores necessários na pior das hipóteses. O NMC, por sua vez, apresenta os maiores resultados necessitando de 33 controladores na pior das hipóteses, reduzindo conforme o aumento do número de passos, atingindo a média de 19 controladores na melhor das hipóteses. O MFC fica entre NME e NMC, apresentando uma média de 14,2 controladores na melhor das hipóteses.

Os resultados apresentados para SYNTH50 corroboram com a afirmação de que uma topologia altamente conectada exigirá um número menor de controladores ao colocar controladores em nós com maior excentricidade. Nesse caso, podemos concluir que, para cenários de carga otimizados, a política NME é recomendada para topologias altamente conectadas. Além disso, para topologias menos conectadas, o MFC apresentará melhores resultados ao planejar a instalação de novos controladores na rede. Além disso, um tempo de implantação com várias etapas de transição pode equilibrar o desempenho entre as políticas selecionadas. No entanto, topologias altamente conectadas ainda se beneficiarão com um número menor de controladores usando a política NME.

## 6.2 Análise dos resultados para carga pesada

Semelhante aos resultados anteriores, a Figura 6.4 mostra os resultados obtidos em relação ao número médio de controladores para o cenário de carga pesada. Na Figura 6.4a, independentemente do número de etapas de transição, o modelo otimizado e o MFC têm

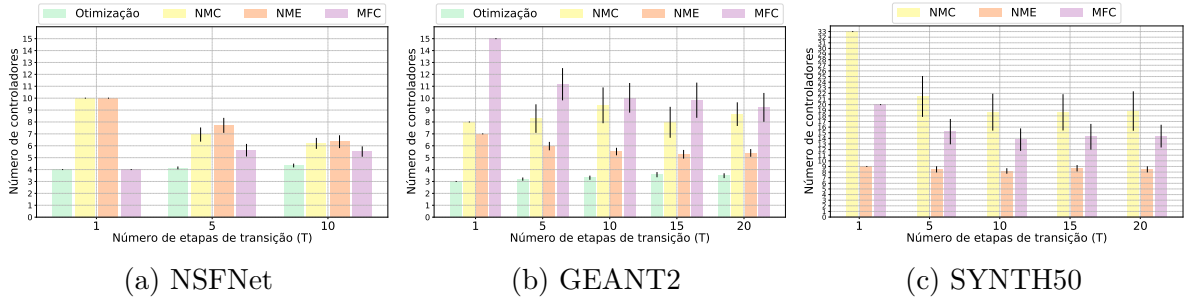


Figura 6.3: Número de controladores nos cenários de carga otimizada.

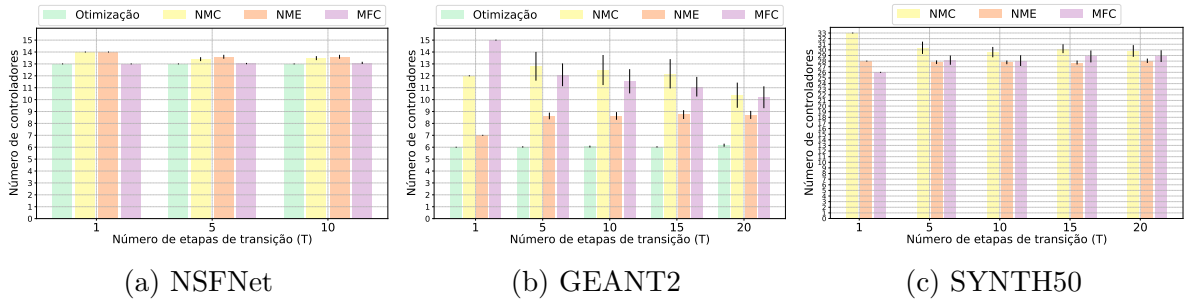


Figura 6.4: Número de controladores nos cenários de carga pesada.

o mesmo número de controladores. NMC e NME, por sua vez, têm quase os mesmos resultados. Como NFSNet apresenta alto tráfego por nó de acordo com o intervalo  $K_n$  na Tabela 6.1, usando a função de carga pesada que aumenta a carga linearmente ao tráfego experimentado, o número médio de controladores é dominado pela restrição de carga. Neste caso, quase todos os *switches* da rede precisam de um controlador dedicado, o que é confirmado pelo valor médio de 13 controladores na melhor das hipóteses pela otimização para uma rede de 14 nós, como NSFNet.

Na Figura 6.4b, é mostrado o número médio de controladores da topologia GEANT2. Dado que GEANT2 é altamente conectada, com pequeno tráfego por nó, seus resultados não são dominados pela restrição de carga, como ocorreu com a rede anterior. Mesmo sob carga pesada, a melhor política é o NME com 8,5 controladores no pior cenário. Por outro lado, a política que requer o maior número de controladores é o MFC com 15 e 10 controladores para o pior (ou seja, 1 etapa de transição) e o melhor caso (ou seja, 20 etapas de transição), respectivamente. O NMC, por sua vez, tem desempenho um pouco melhor do que o MFC, mas apresenta o mesmo comportamento nas etapas de transição. Conforme analisado anteriormente para o cenário de carga otimizada, o mesmo ocorre para o cenário de carga pesada, onde o maior número de transições serve para balancear os resultados das heurísticas. No entanto, o NME ainda é uma heurística melhor apesar do número de etapas de transição, o que também é confirmado pela análise de variabilidade, onde os intervalos de confiança não são sobrepostos.

A próxima análise diz respeito à topologia SYNTH50 para as heurísticas propostas, Figura 6.4c. A política NMC resultou no maior número de controladores, enquanto NME foi a menor, com 28 controladores em média. A diferença no número de controladores entre as políticas foi mais significativa no cenário de carga otimizada da Figura 6.3 do que na carga pesada exibida na Figura 6.4. Apesar de SYNTH50 ser uma rede altamente conectada, seu alto tráfego por nó também gerou carga alta, tornando o número médio de controladores mais sensível à restrição de carga. Isso significa que os resultados são quase os mesmos, apesar da heurística adotada ou do número de transições consideradas.

Como observação final, os cenários de carga pesada representam um ambiente de rede muito mal controlado, a partir do qual cada pacote de dados que ocorre cria uma mensagem *packet\_in* para o controlador. O custo de sinalização é igual ao tráfego de dados, o que é muito improvável de ocorrer na realidade. Ao mesmo tempo, uma vez que não estamos avaliando nem otimizando o recurso de controle da rede SDN, nossos resultados cobrem esse tipo de cenário para completude.

# Capítulo 7

## Conclusões e Trabalho Futuro

Nesta monografia, propomos um modelo de otimização para o problema de posicionamento do controlador para a transição de uma rede legada para híbrida até que se torne totalmente SDN. Dado que a solução para o problema de otimização requer o conhecimento total de cada etapa de transição de uma rede, propomos três políticas alternativas baseadas em invariante de grafos com diferentes heurísticas para selecionar o melhor nó candidato a hospedar um controlador, são elas (i) Nó Mais Conectado; (ii) Nó com Maior Excentricidade; e (iii) Nó com Maior Fluxo de Carga. Os resultados mostraram que o resultado ideal tem pouco impacto durante a transição da rede legada para SDN. No entanto, os resultados das políticas têm grande influência do número total de transições, permitindo que o número de controladores seja balanceado ao final de um grande número de transições. Além disso, detectamos que topologias altamente conectadas podem reduzir muito o número de controladores necessários quando planejados com a política de nó com maior excentricidade. Finalmente, as topologias menos conectadas requerem menos controladores quando planejadas com o uso da política de nó com maior fluxo de carga usando um pequeno tempo de implementação, ou seja, poucas etapas de transição.

Dadas essas informações, reconhecemos algumas limitações ao amplo uso de nosso modelo em cenários de rede reais. Primeiro, um controlador não pode ser migrado para outro local depois de colocado. Sabemos que pode ser mais barato mover um controlador para outro local em vez de comprar um novo. No entanto, como nos concentramos em uma rede operada por equipes de gestão diferentes e independentes, instalar um controlador em um departamento e, em seguida, revogá-lo pode gerar conflito entre eles. Em segundo lugar, assumimos que o controlador pode ser instalado no local escolhido. Às vezes, não há infraestrutura ou orçamento para colocar e manter esse equipamento em cada local em que há um *switch*. Finalmente, a migração da rede é aleatória. Algumas empresas podem traçar um plano para atualizar sua rede para SDN. Por outro lado, por estarmos considerando redes gerenciadas por equipes independentes, o momento em que



um departamento atualiza seus *switches* depende de diversas variáveis, como necessidades, orçamento, infraestrutura e burocracia. Portanto, decidimos que uma migração aleatória de *switch* se encaixaria melhor em nosso caso de estudo.

Como trabalho futuro, sugerimos explorar um modelo de otimização em uma rede centralizada, onde o plano de migração da rede faz parte do resultado. Otimização com objetivos diferentes, como resiliência e balanceamento de carga, também pode ser explorada. Além disso, a função *mapear\_switches* do Algoritmo 1 pode ser melhorada para resultados otimizados em termos de balanceamento de carga.

# Referências

- [1] Sezer, Sakir, Sandra Scott-Hayward, Pushpinder Chouhan, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Marc Miller e Navneet Rao: *Are we ready for SDN? Implementation challenges for software-defined networks*. IEEE Communications Magazine, páginas 36–43, 2013. 1
- [2] Binlun, Jason Ng, Tan Saw Chin, Lee Ching Kwang, Zulfadzli Yusoff e Rizaluddin Kaspin: *Challenges and Direction of Hybrid SDN Migration in ISP networks*. Em *2018 IEEE International Conference on Electronics and Communication Engineering, ICECE 2018*, páginas 60–64, 2019. 1
- [3] Marotta, Marcelo A., Maicon Kist, Juliano A. Wickboldt, Lisandro Z. Granville, Juergen Rochol e Cristiano B. Both: *Design considerations for software-defined wireless networking in heterogeneous cloud radio access networks*. Springer Journal of Internet Services and Applications, página 18, 2018. 1
- [4] Amin, Rashid, Martin Reisslein e Nadir Shah: *Hybrid SDN Networks: A Survey of Existing Approaches*. IEEE Communications Surveys & Tutorials, páginas 3259–3306, 2018. 1
- [5] Kreutz, Diego, Fernando M.V. Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky e Steve Uhlig: *Software-defined networking: A comprehensive survey*. Proceedings of the IEEE, páginas 14–76, 2015. 1
- [6] Qin, Qiaofeng, Konstantinos Poularakis, George Iosifidis e Leandros Tassiulas: *SDN Controller Placement at the Edge: Optimizing Delay and Overheads*. Em *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, páginas 684–692, 2018. 1
- [7] Das, Tamal e Mohan Gurusamy: *INCEPT: INcremental ControllEr PlacemenT in Software Defined Networks*. Em *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, volume 2018-July, páginas 1–6, 2018. 2, 4
- [8] Das, Tamal, Vignesh Sridharan e Mohan Gurusamy: *A Survey on Controller Placement in SDN*. IEEE Communications Surveys & Tutorials, páginas 472–503, 2020. 2, 6
- [9] Das, Tamal e Mohan Gurusamy: *Resilient Controller Placement in Hybrid SDN/Legacy Networks*. Em *2018 IEEE Global Communications Conference (GLOBECOM)*, páginas 1–7, 2018. 2, 4

- [10] Xu, Hongli, He Huang, Shigang Chen, Gongming Zhao e Liusheng Huang: *Achieving High Scalability Through Hybrid Switching in Software-Defined Networking*. IEEE/ACM Transactions on Networking, páginas 618–632, 2018. 2
- [11] Fakhteh, Amir Hossein, Vahid Sattari-Naeini e Hamid Reza Naji: *Increasing The Network Control Ability and Flexibility In Incremental Switch Deployment for Hybrid Software-Defined Networks*. Em *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, páginas 263–268, 2019. 3
- [12] Csikor, Levente, Mark Szalay, Gabor Retvari, Gergely Pongracz, Dimitrios P. Pezaros e Laszlo Toka: *Transition to SDN is HARMLESS: Hybrid Architecture for Migrating Legacy Ethernet Switches to SDN*. IEEE/ACM Transactions on Networking, páginas 275–288, 2020. 3
- [13] Ding, Damu, Marco Savi, Gianni Antichi e Domenico Siracusa: *An Incrementally-Deployable P4-Enabled Architecture for Network-Wide Heavy-Hitter Detection*. IEEE Transactions on Network and Service Management, páginas 75–88, 2020. 3
- [14] Cabellos, Albert: *Knowledge-defined networking training datasets*. <https://knowledgedefinednetworking.org/>. 6, 14, 15
- [15] Rahman, Shams ur, Geon Hwan Kim, You Ze Cho e Ajmal Khan: *Deployment of an sdn-based uav network: Controller placement and tradeoff between control overhead and delay*. Em *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, páginas 1290–1292, 2017. 11
- [16] Pedreno-Manresa, Jose Juan, Jose Luis Izquierdo-Zaragoza e Pablo Pavon-Marino: *Joint fault tolerant and latency-aware design of multilayer optical networks*. páginas 1–6, 2016. 15
- [17] Tom, C., H. Yu, K. Li e H. Qi: *An active controller selection scheme for minimizing packet-in processing latency in sdn*. Em *Security and Communication Networks*, 2019. 15
- [18] Yao, L., P. Hong e W. Zhou: *Evaluating the controller capacity in software defined networking*. Em *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, páginas 1–6, 2014. 15
- [19] Filali, A., A. Kobbane, M. Elmachkour e S. Cherkaoui: *Sdn controller assignment and load balancing with minimum quota of processing capacity*. Em *2018 IEEE International Conference on Communications (ICC)*, páginas 1–6, 2018. 15
- [20] Zhao, G., L. Huang, Z. Yu, H. Xu e P. Wang: *On the effect of flow table size and controller capacity on sdn network throughput*. Em *2017 IEEE International Conference on Communications (ICC)*, páginas 1–6, 2017. 15
- [21] Pranata, Alif Akbar, Tae Soo Jun e Dong Seong Kim: *Overhead reduction scheme for sdn-based data center networks*. Computer Standards Interfaces, páginas 1 – 15, 2019. 15

# Apêndice A

## Artigo Publicado

Esta monografia é um artigo submetido e publicado na *IFIP/IEEE International Symposium on Integrated Network Management* (IM 2021).

- Título: On the Transition of Legacy Networks to SDN - An Analysis on the Impact of Deployment Time, Number, and Location of Controllers.
  - Autores: PONTES, D. F. ; CAETANO, M. F. ; FILHO, G. P. ; GRANVILLE, L. Z. ; MAROTTA, M. A.
  - Conferência: IFIP/IEEE International Symposium on Integrated Network Management.
  - Data: Maio, 2021.