

Cobertura mínima de vértices na migração para SDN

Eduardo Augusto Klosowski¹, Adriano Fiorese¹

¹ Departamento de Ciências da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)
Joinville – SC – Brasil

eduardo.klosowski@edu.udesc.br, adriano.fiorese@udesc.br

Resumo. *Este trabalho visa minimizar a quantidade de equipamentos substituídos em uma migração de rede legada para uma rede híbrida, aumentando assim a viabilidade financeira da migração, podendo usufruir de funcionalidades possibilitadas pelo paradigma de Redes Definidas por Software (SDN). Para isso são verificados diferentes métodos para se encontrar soluções para a cobertura mínima de vértices, que é um problema NP-difícil.*

1. Introdução

O paradigma de Redes Definidas por Software (*Software-Defined Networking*, SDN) oferece uma abordagem heterodoxa para o funcionamento das redes de computadores [Kreutz et al. 2015]. Ela facilita o uso de novas tecnologias de rede, como utilização mais eficiente dos enlaces de rede e flexibilidade de configuração. Porém muitas empresas possuem uma infraestrutura de rede legada (uma rede que não é SDN), necessitando muitas vezes da troca dos equipamentos para suportar protocolos utilizados em redes SDN, como o OpenFlow [McKeown et al. 2008]. Na construção de uma nova infraestrutura, além dos custos dos novos equipamentos, equipamentos legados, ainda em funcionamento, podem ser desmobilizados, afetando o *capital expenditure* (CAPEX) da empresa, o que em última análise, poderia inviabilizar financeiramente o projeto de migração.

Sabendo que é possível utilizar equipamentos SDN e legados em uma mesma rede, tornando-a nesse sentido, híbrida, pergunta-se como minimizar a quantidade de equipamentos que seria necessário trocar para implementar uma solução SDN, reduzindo assim seu custo de implementação.

Este trabalho procura estudar como modelar a migração de uma rede tradicional legada para uma rede híbrida com *switches* OpenFlow. Para tal, discute-se diversos algoritmos que busquem a menor quantidade possível de dispositivos que são necessários substituir para implementar a mesma rede, porém compatível com funcionalidades baseadas em SDN.

2. Modelagem do problema

Uma rede pode ser representada como um grafo, onde os vértices representam os *hosts* e dispositivos de encaminhamento, enquanto as arestas seus enlaces. Porém deve-se manter apenas as arestas onde alguma regra SDN será necessária. Isso deve ocorrer conforme as restrições das funcionalidades SDN que serão aplicadas, podendo incluir todos os enlaces ou apenas parte deles, por exemplo, para monitorar toda a rede, ou apenas enlaces com dispositivos finais, para aplicar regras de acesso, respectivamente. Os vértices que ficarem sem arestas podem ser removidos.

A Figura 1a ilustra uma rede a ser migrada, sendo a Figura 1b o grafo resultante de toda a rede, enquanto a Figura 1c inclui apenas os enlaces com dispositivos finais.

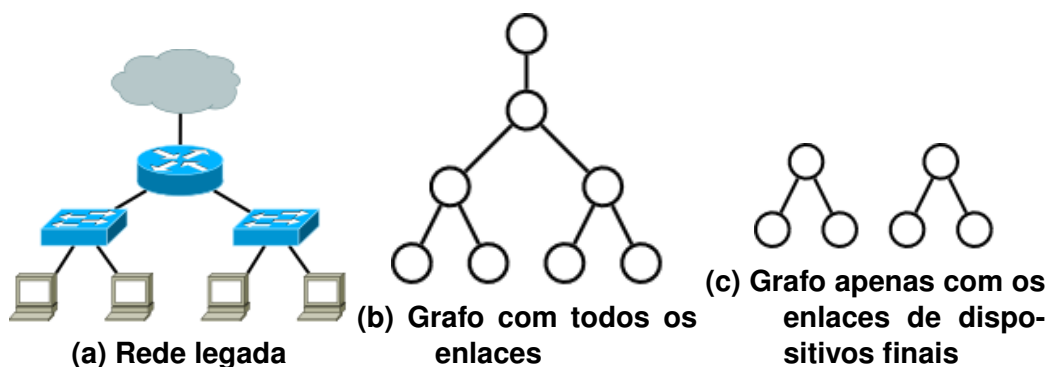


Figura 1. Grafo da rede

Caso exista algum dispositivo de encaminhamento que deve ser migrado para SDN por alguma restrição do projeto, seu vértice deve ser marcado como um dispositivo que será substituído e removido do grafo com suas arestas. Caso algum dispositivo não possa ser migrado para SDN, seu vértice também deve ser removido do grafo com suas arestas. Da mesma forma que dispositivos que já possuem o suporte a SDN, podem ser removidos do grafo, uma vez que se pode utilizar o suporte a SDN do dispositivo, cobrindo seus enlaces, sem aumentar os custos da migração.

Assim, dado o grafo $G = (V, A)$, para identificar os dispositivos que teriam que ser migrados, bastaria encontrar um conjunto de vértices $C \subseteq V$, de tal forma que ao menos um dos vértices de cada aresta A esteja contido em C . Porém, também deseja-se que C seja o menor conjunto possível, visto que isso reduziria a quantidade de dispositivos trocados. Esse problema é conhecido como cobertura mínima de vértices (*Minimum Vertex Cover*, MVC) [Garey and Johnson 1990]. Formas de encontrar o conjunto C serão discutidas posteriormente na Seção 3.

No caso particular em que G é um grafo desconexo, cada parte conexa pode ser processada paralelamente, e posteriormente, feita a união dos conjuntos C resultantes.

3. Resolvendo a cobertura mínima de vértices

O problema da cobertura mínima de vértices é um problema NP-difícil, enquanto sua versão de decisão é NP-completo [Garey and Johnson 1990]. Sendo assim, é possível obter uma resposta válida (conjunto de vértices que cubra todas as arestas), porém dificilmente se obterá a resposta ótima (menor conjunto possível de vértices) para problemas maiores em um curto espaço de tempo, visto que o espaço de busca é exponencial ao número de vértices do grafo.

3.1. Algoritmo guloso

Uma forma simples de se obter uma resposta válida, mas não necessariamente a resposta ótima, é por meio de um algoritmo guloso. Sua lógica consiste em ir adicionando a um conjunto C vértices com a maior quantidade de arestas ainda não cobertas por C , até que todas as arestas tenham sido cobertas.

Embora essa lógica possa encontrar a resposta ótima para alguns tipos de grafos, como a classe *p-hat* do *benchmark* DIMACS, ela não garante que a resposta encontrada será a ótima em todos os casos [Richter et al. 2007].

3.2. Heurísticas

Algoritmos heurísticos executam em menor tempo, quando comparados aos algoritmos que testam todas as possibilidades de solução. Normalmente encontrando uma solução melhor que o algoritmo guloso, mesmo que possam excluir a possibilidade de encontrar a solução ótima no processo.

3.2.1. Cover Edges Randomly

O *Cover Edges Randomly* (COVER) é uma heurística que usa técnicas aplicadas na busca de valores para satisfazer uma expressão booliana (*boolean satisfiability problem* ou SAT) ao MVC [Richter et al. 2007]. Ele é um algoritmo *Stochastic Local Search* (SLS) que a partir de uma resposta candidata inicial C com k vértices, construída de forma gulosa, troca um vértice do conjunto C a cada passo, transformando-a em respostas vizinhas, até encontrar uma resposta que seja a cobertura de vértices ou atingir o número máximo de passos [Richter et al. 2007].

Os vértices trocados são escolhidos por meio de heurísticas (critérios) baseadas em o quão difícil é cobrir as arestas, mas incluindo também uma parte de aleatoriedade para escapar de ótimos locais.

Porém, o COVER procura apenas respostas com k vértices. Caso k não seja conhecido, a opção é usar o COVER-I, onde inicia-se k com a quantidade de vértices encontrada em uma solução gulosa, decrementando-a em 1 até que não se encontre uma solução e retorna-se o último conjunto gerado.

3.2.2. Edge Weighting Local Search

O *Edge Weighting Local Search* (EWLS) é uma heurística que busca encontrar uma cobertura parcial de vértices otimizada e expandi-la para uma cobertura total, onde na pior das hipóteses seria necessário adicionar um vértice para cada aresta não coberta pela cobertura parcial [Cai et al. 2010]. Cada solução é comparada com a melhor solução já encontrada, substituindo-a sempre que tiver uma quantidade menor de vértices.

A cobertura parcial de vértices é delimitada por uma quantidade *delta* de vértices retirados de uma cobertura completa (inicialmente construída de forma gulosa), onde quanto maior o *delta*, maior o espaço de busca. Aumentando o espaço de busca, pode-se encontrar melhores soluções, porém exige mais recursos e não garante a otimização da cobertura ao expandi-la. A cada passo, o EWLS também troca um vértice do conjunto C , de forma semelhantes ao COVER, porém aplica a forma aleatória apenas quando encontra um ótimo local.

4. Considerações

Observou-se que, em uma migração de rede legada para híbrida, é possível encontrar a quantidade de dispositivos que precisam ser substituídos, porém para minimizar os custos

é necessário resolver o problema da cobertura mínima de vértices.

Entretanto, ele é um problema NP-difícil, o que dificulta encontrar a solução ótima. Desta forma é necessário o uso de alguma técnica para encontrar boas soluções em tempo hábil, como as heurísticas COVER e EWLS. Porém, em seus artigos seminais são discutidos apenas seus desempenhos em testes de *benchmark*, onde se conhece previamente a quantidade de vértices da solução no caso do COVER, ou o melhor parâmetro de *delta* para o EWLS, encontrando a solução já conhecida. Não são discutidos quais seriam os melhores parâmetros para grafos desconhecidos, e como encontrar um equilíbrio para a quantidade de passos, visto que quanto maior esse valor, maior a possibilidade de se encontrar uma melhor solução, porém maior o tempo para a execução das heurísticas.

5. Continuidade

A continuidade deste trabalho poderia buscar o ponto onde uma solução de força bruta deixaria de retornar uma resposta em tempo hábil para o MVC, visto que é possível encontrar a solução ótima para pequenas migrações. Para isso, é necessário dispor das características do grafo gerado por diferentes redes de computadores, de forma que se possa alterar o tamanho do grafo para, reexecutando o algoritmo de força bruta, identificar em que ponto o tempo de resposta do algoritmo se torna inviável.

Essas características poderiam ser extraídas de diferentes topologias, como redes empresariais, *datacenters* e WANS, podendo identificar diferentes tipos, gerando grafos de diferentes complexidades. Uma vez em posse dessa informação, poder-se-ia analisar os testes de *benchmark* presentes nos trabalhos [Richter et al. 2007, Cai et al. 2010] em busca de classes de testes que poderiam representar essas diferentes topologias, identificando assim também quais heurísticas apresentam melhor desempenho para essas classes de grafos, além de guiar a busca de quais seriam bons parâmetros para a utilização das heurísticas para se resolver esses problemas.

Referências

- Cai, S., Su, K., and Chen, Q. (2010). Ewls: A new local search for minimum vertex cover. In *AAAI*.
- Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.
- Richter, S., Helmert, M., and Gretton, C. (2007). A stochastic local search approach to vertex cover. In *Annual Conference on Artificial Intelligence*, pages 412–426. Springer.