

Geração Aleatória de Topologias de Rede em SDN

Willyan C. Silva¹, Brivaldo A. S. Junior¹

¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 549 – 79.070-900 – Campo Grande – MS – Brasil

willyan.silva@aluno.ufms.br, brivaldo@facom.ufms.br

Abstract. *Researchs in computer networks is intimately connected to topology generators, because these generators provide quickly and artificial simulation of real network topologies. This paper presents a new local area network topology generator based on statistical probability distributions applied in undirected acyclic graphs. The main objective of this work is generate simulated topologies on software defined networking (SDN) to allowing faster experimentations on this new network approach.*

Resumo. *O campo de pesquisas em redes de computadores está intimamente ligado com o uso de geradores de topologias. Isso porque esses geradores fornecem de forma rápida e artificial a reprodução de redes semelhantes a topologia real. Este trabalho apresenta um novo gerador de topologias de rede locais, produzido com base em distribuições probabilísticas aplicadas em grafos acíclicos não dirigidos. O objetivo final deste trabalho é gerar topologias simuladas em ambientes de redes definidas por softwares (SDN) para permitir uma experimentação mais rápida nesta nova proposta de rede.*

1. Introdução

Durante o processo de criação de novos protocolos de rede são realizados exaustivos testes e simulações com intuito de avaliar sua viabilidade em termos de eficiência e desempenho. De modo geral, essas avaliações utilizam-se de algumas características da rede para comparar e determinar o comportamento desses protocolos. A escolha certa das características é fundamental, dado que elas devem demonstrar as propriedades encontradas em uma estrutura de rede pois, caso contrário, não se pode tirar conclusões dos resultados obtidos.

Entretanto, possuir conhecimento básico elementar das topologias pode não ser uma tarefa tão trivial, haja visto que estão em constante evolução e que nem sempre essas informações estão disponíveis pública. Em razão disso, os pesquisadores da área de redes utilizam geradores de topologias para realizarem simulações e testes de novos protocolos e aplicações.

O primeiro gerador de topologias amplamente utilizado em simulações foi desenvolvido por [Waxman 1988]. Criado a partir de uma variação do modelo clássico de grafos randômicos de Erdos-Renyi [Faloutsos et al. 1999], no qual uma aresta é

criada com base na distância Euclidiana entre dois nós. Dentre os vários geradores, os mais conhecidos são: GT-ITM [Calvert et al. 1997], Tiers [Doar 1996] e BRITE [Medina et al. 2001]. O principal problema observado nesses geradores é que eles são voltados para concepção de topologias modeladas para a Internet, e não representam, de forma simples e direta as características identificadas em redes locais. Outros dois modelos muito utilizados nessa área de pesquisa são as distribuições baseadas em grau [Tangmunarunkit et al. 2002], usadas para geração de redes em nível de roteador (*Router Level - RL*) e sistemas autônomos (*Autonomous System - AS*). A principal diferença entre os dois modelos encontra-se na quantidade de conexões que cada nó possui.

Este trabalho apresenta o desenvolvimento de uma ferramenta geradora de topologias de redes locais (*Local Area Network - LAN*) aleatórias para simulação de topologias de rede em Redes Definidas por Software (*Software Defined Networking - SDN*) [Koldehofe et al. 2012], e que forneça uma rede semelhante a topologia real. Uma característica interessante que essa topologia deve conter é a quantidade de conexões vinculadas a um nó/vértice, *i.e.*, a qual classe o nó pertence ou o grau dele (ex.: *host*, *switch*, roteador).

A necessidade de uma nova ferramenta geradora é dada pelo fato dos atuais geradores serem especialistas na geração de topologias voltadas ao modelo atual. A geração das topologias é realizada com base no modelo aleatório de Waxman e inserção aleatória de nós usando modelos probabilísticos distintos. Além disso, o pesquisador tem a opção de escolher a distribuição estatística que melhor se encaixe as suas necessidades, como: discreta, normal, uniforme, *poisson*.

O trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta modelos probabilísticos aleatórios na geração de números utilizando as distribuições normal e exponencial. A Seção 4 apresenta a arquitetura de desenvolvimento utilizando grafos e o uso de SDN para simulações, assim como os resultados obtidos em experimentações. E, finalmente, a Seção 5 apresenta as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Um dos mais conhecidos geradores de topologia, o BRITE [Medina et al. 2001], foi desenvolvido com intuito de ser um gerador de topologias universais. Para atingir esse objeto ele faz uso de oito classes com modelos diferentes, em que o modelo possui um método gerador responsável por devolver um grafo da topologia gerada. Essas classes são divididas em duas categorias. A primeira categoria, contém modelos *ad-hoc*, foi construída com base em palpites e é composta pelos modelos de Waxman, GT-ITM e Tiers. A segunda categoria inclui modelos construídos usando medições (por exemplo, power-law e *outdegrees*) e é composta por modelos como Inet e PLRG. As topologias por ele geradas podem ser diretamente usadas nos simuladores NS (*Network Simulator*) [Riley 2003] e SSF (*Scalable Simulation Framework*) [Nicol et al. 2003].

[Tangmunarunkit et al. 2002] definiram, usando várias métricas de topologias, qual gerador de topologias melhor representava as estruturas em larga escala da Internet. Os resultados de sua pesquisa foram avaliados com base em três métricas: expansão, resiliência e distorção. A expansão tem aspecto fundamental para definir o tamanho da vizinhança com base no número de sites que podem ser alcançados atravessando h saltos. A resiliência mede a robustez do grafo se alguma aresta falhar, como no caso de uma bipartição. A distorção leva em consideração qualquer árvore de expansão (*spanning tree*) T em um grafo G e calcula a distância média em T e quaisquer dois vértices que compartilhem uma aresta em G . Em sua pesquisa observou-se que os geradores baseados em distribuições de grau são os que mais se assemelham a natureza hierárquica da Internet.

3. Modelos Aleatórios

A geração de números e variáveis aleatórias é essencial em qualquer programa de simulação, pois possibilita a criação e o teste de vários cenários distintos. A motivação para o uso da aleatoriedade deve-se ao fato de tentar reproduzir em um ambiente controlado algo que possa acontecer no mundo real. Para simulações em ambientes computacionais, utilizam-se números pseudoaleatórios baseados em uma distribuição probabilística, tal como a distribuição Gamma, a de Poisson, a Uniforme, entre outras. Esses geradores podem produzir uma mesma sequência se forem inicializados com a mesma semente (*seed*). Além disso, um gerador aleatório é bom se ele possui uma alta periodicidade.

Os modelos probabilísticos possuem duas componentes: o espaço amostral (S) que é o conjunto de eventos possíveis em um experimento e a probabilidade de eventos (P) que é a determinação da quantidade que um certo evento ocorra. Neste trabalho foram utilizados dois modelos probabilísticos aleatórios: distribuição normal e exponencial.

3.1. Distribuição Normal

A distribuição Normal é uma das mais importantes distribuições de probabilidade Estatística. Ela também é conhecida como distribuição de Gauss. A distribuição de probabilidade de uma variável aleatória x é chamada normal se tem como função densidade de probabilidade:

$$P(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (1)$$

O parâmetro μ é a média ou expectativa da distribuição. O parâmetro σ é o seu desvio padrão, e sua variância é σ^2 .

3.2. Distribuição Exponencial

A distribuição Exponencial pertence a classe contínua de uma variável aleatória x que é definida pela densidade. O único parâmetro da distribuição é a média, representada por

λ . Sua função de densidade de probabilidade é expressa por:

$$P(x) = \begin{cases} \lambda - e^{-\lambda x} & , x \geq 0, \\ 0 & , x < 0. \end{cases} \quad (2)$$

A distribuição Exponencial é amplamente utilizada em sistemas de cálculo como teoria de filas e confiabilidade. Também é considerada uma das mais simples em termos matemáticos.

4. Implementação

Este trabalho tem por objetivo, preencher o vazio na área de geração aleatória de topologias de redes locais para simulação de novas tecnologias em redes definidas por *software*, uma vez que os geradores atuais são especializados na criação de topologias voltadas para a Internet. Isso gera certa dificuldade para os pesquisadores que trabalham com redes locais, porque as propriedades entre cada uma das topologias é bem diferente.

A ferramenta foi desenvolvida em Python por dois motivos. Primeiro pelo fato de ser uma linguagem amplamente utilizada nos dias atuais, que fornece segurança e facilidade para o desenvolvimento (boa comunidade e documentação). Segundo, porque o Mininet [Lantz et al. 2010] (ferramenta empregada no ambiente de simulação SDN) possui API's (*Application Programming Interface*) que trabalham com C++ e Python. O uso da ferramenta será discutido posteriormente. Ademais, usou-se uma estrutura de grafos acíclicos não dirigidos para construir a topologia de rede, dado que essas propriedades atendem as características de uma rede local em um baixo nível de abstração.

Para facilitar a visualização das informações produzidas pelo programa, foi utilizado o pacote PyGraphviz, derivado do pacote Graphviz e semelhante NetworkX (biblioteca de manipulação de grafos). A adoção do PyGraphviz foi determinada pelo fato de não haver necessidade de uma ferramenta tão robusta para a manipulação do grafo. Com isso, foi possível criar uma representação visual dos grafos gerados.

SDN (Rede Definida por Software) é um novo paradigma de rede que separa o plano de dados do plano de controle, retirando a inteligência dos dispositivos de rede que passam a ser apenas encaminhadores de pacotes, sua arquitetura pode ser vista na Figura 1. Além disso, o plano de controle é delegado a um controlador que passa a orquestrar a rede. Essa perspectiva proporciona novas possibilidades para o gerenciamento [Kim and Feamster 2013] de rede e métodos de configuração. O protocolo *OpenFlow* [McKeown et al. 2008], foi a primeira interface padrão projetada especificamente para SDN e que já está sendo implantada em uma variedade de redes e produtos baseados em *hardware* ou *software* [Hoelzle 2012] [Higginbotham 2012]. Por ser um paradigma recente, muitos mecanismos atuais precisam ser reimplementados.

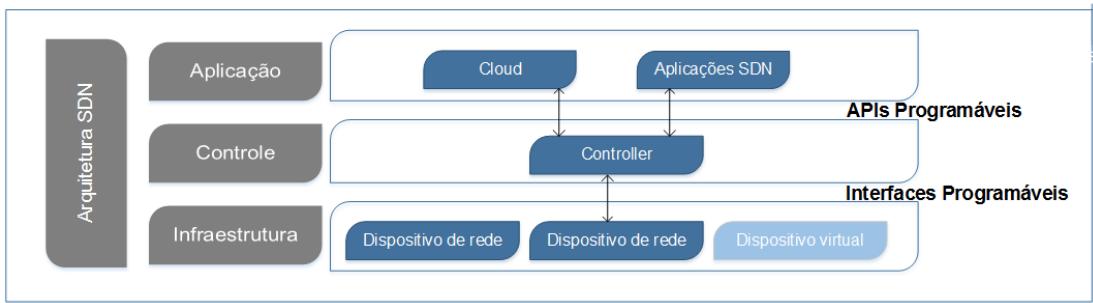


Figura 1. Arquiterura de uma Rede Definida por Software.

Com isso surge o Mininet, que é uma plataforma de simulação de rede muito útil para testar aplicações para SDN. Seu principal objetivo é fornecer uma ferramenta leve que possibilite uma rápida prototipagem para criar, interagir, personalizar e compartilhar uma rede, assim como sua execução em hardware. O Mininet é distribuído em forma de máquinas virtuais (VM) ou em pacotes para instalação nativa. Com ele é possível simular importantes características da rede como: largura de banda, atraso, taxa de perda de pacotes, etc. Além disso, é possível também executar testes de *ping* e *dump* das conexões para avaliações em tempo real.

A ferramenta desenvolvida está divida em duas partes: gerador e classe que implementa a aleatoriedade de acordo com o modelo estatístico. Na Figura 2 pode-se observar essa separação e o fluxo de execução da ferramenta. A primeira parte da ferramenta é composta por um único arquivo em que é realizada a leitura dos parâmetros de entrada: quantidade de *hosts*, *switches* e o número de portas que cada *switch* possui. É importante ressaltar que o sistema não realiza validação do conjunto de entrada, seja ela do tipo de entrada ou se o número de portas é suficiente para a quantidade de nós/vértices (*hosts* e *switches*). A segunda parte é composta por duas classes: Normal e Exponencial. Após a leitura dos parâmetros, duas instâncias são criadas tomando como argumento as informações da leitura. Então, chama-se o método de criação de topologias de rede aleatórias.

As classes tem funcionamento similar, diferenciando-se apenas na distribuição estatística que é utilizada para aleatoriedade. Inicialmente são criados os conjuntos de nós e um grafo nulo (sem nenhum nó). Posteriormente faz-se a ligação dos *switches* e logo em seguida a conexão dos *hosts*, adicionando essas informações no grafo. A biblioteca *random* do Python é quem fornece as distribuições de probabilidade utilizadas pela ferramenta. Na classe Normal o gerador de aleatoriedades recebe como argumento os parâmetros "*mu*" (μ) que é a média e "*sigma*" (σ) que é o desvio padrão. Na classe exponencial o gerador de aleatoriedades recebe um único argumento "*lambda*" (λ) que é o valor 1.0 dividido pela média desejada.

Após a conclusão das etapas anteriores, o grafo passa por um processo de coesão, ou seja, é verificado se algum vértice ficou sem adjacência. Caso isso ocorra é feito uma nova tentativa de conexão com um *switch* que tenha portas disponíveis. Se mesmo após

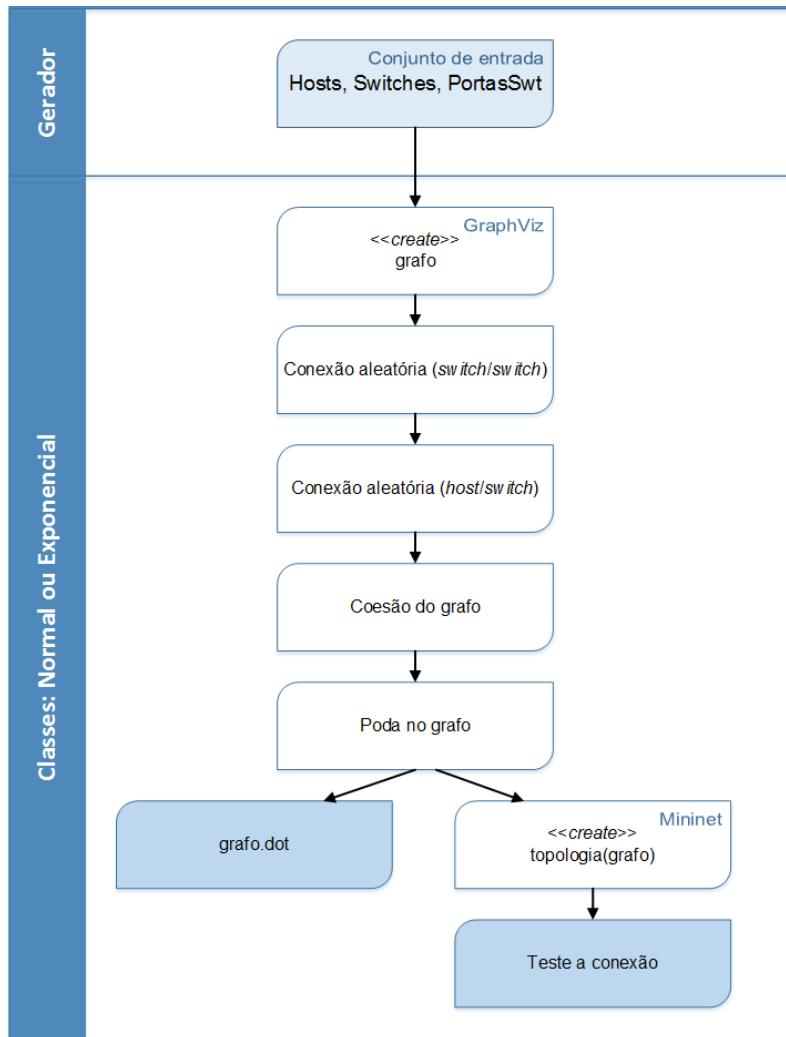


Figura 2. Fluxo de execução da ferramenta na geração e simulação de topologias de rede.

executar o passo anterior ainda houver vértices desconexos, então eles são removidos do grafo. Finalmente o grafo da topologia está concluído e então ele é desenhado em um arquivo de saída.

Por fim as informações do grafo são coletadas para a construção da rede virtual no Mininet. Para melhor visualizar a rede criada, dois testes são realizados: *dump* das conexões de cada *host* e teste de acessibilidade entre todos os *hosts* por meio de *ping*. Então o Mininet é encerrado e a execução da ferramenta chega ao fim.

Para melhor compreensão, dois experimentos foram executados na ferramenta. O primeiro recebeu um conjunto de entrada contendo dez *hosts* e cinco *switches* com cinco portas cada. Os grafos gerados como resultado dos experimentos estão representados na Figura 3 (distribuição normal) e Figura 4 (distribuição exponencial). O segundo experimento teve como entrada 50 *hosts*, cinco *switches* com 12 portas cada, utilizando as mesmas distribuições aleatórias citadas e que podem ser visualizadas na Figura 5

(distribuição normal) e Figura 6 (distribuição exponencial).

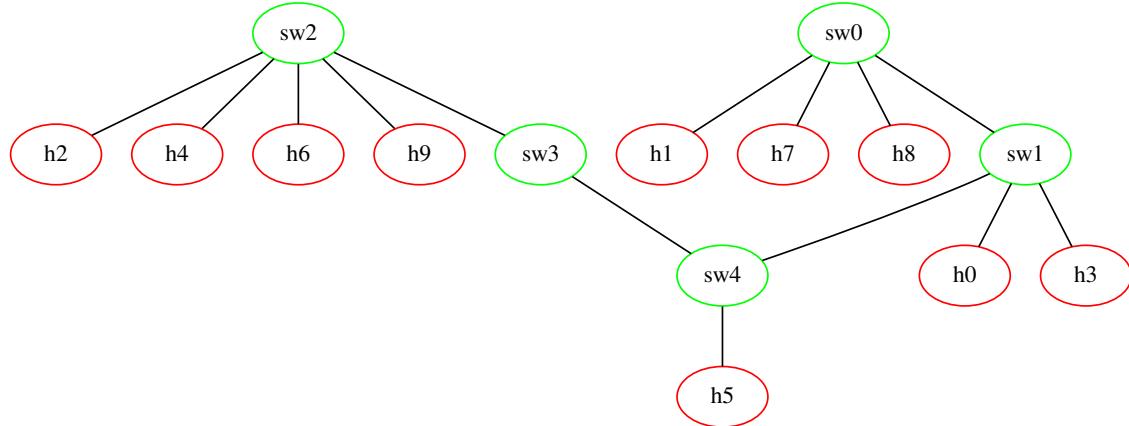


Figura 3. Aleatoriedade Normal (10 hosts, 5 switches, 5 portas/switch).

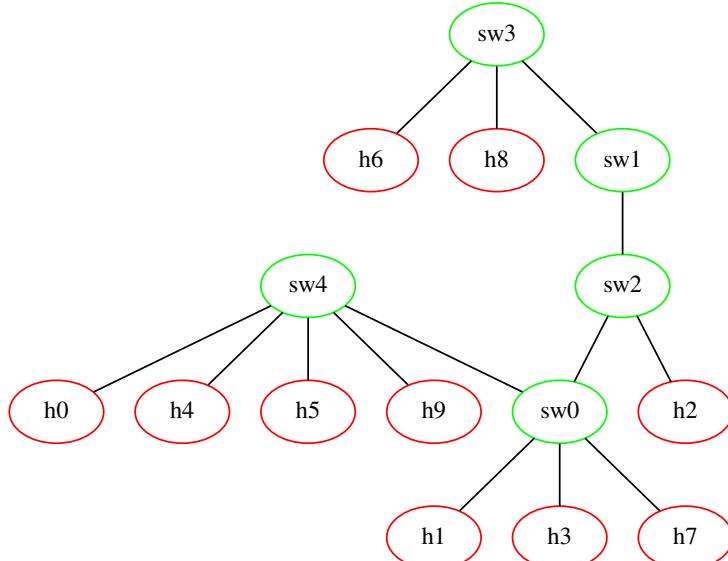


Figura 4. Aleatoriedade Exponencial (10 hosts, 5 switches, 5 portas/switch).

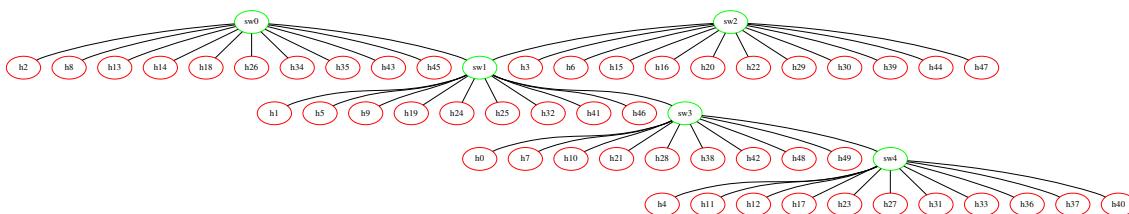


Figura 5. Aleatoriedade Normal (50 hosts, 5 switches, 12 portas/switch).

Após a geração das topologias citadas acima, cada uma delas passou pela etapa de simulação no ambiente de SDN provido pelo Mininet. Além do processo de construção da rede, as redes geradas foram submetidas a testes de conexão *all-to-all*

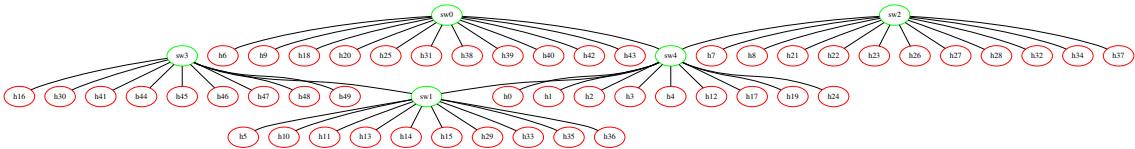


Figura 6. Aleatoriedade Exponencial (50 hosts, 5 switches, 12 portas/switch).

(teste que dispara pacotes ICMP de cada *host* para todos os outros. Durante as simulações não ocorreram perdas de pacotes e todos os ativos de rede conseguiram comunicar entre si.

5. Conclusão

A ferramenta desenvolvida mostrou-se capaz de produzir bons resultados. Contudo, os resultados mostraram que apesar do uso de duas diferentes distribuições probabilísticas para criação dos grafos não foi possível observar diferenças significativas nos grafos gerados. Nos testes realizados também foi possível gerar topologias com quase 1000 nós, contudo leva-se um tempo considerável na etapa de simulação. A média utilizada pelo modelo probabilístico exponencial deve ser ajustada quanto maior for a topologia desejada, pois o conjunto de eventos pode não ser compatível com o espaço amostral provocando *loop* na ferramenta.

O Mininet é uma ótima ferramenta que proporciona comodidade, realismo, baixo custo (pode ser executado praticamente em qualquer computador) e fácil instalação. Contudo ele possui limitações de simulação, sendo capaz de simular cerca de no máximo 2000 nós e que demanda tempo de execução relativamente oneroso.

O SDN apesar de ser um paradigma novo ele demonstra ser um campo muito promissor na área de redes de computadores, tanto para pesquisa quanto no âmbito comercial. Um fator que pode vir a impulsionar a adoção dessa tecnologia é o fato do escopo de aplicação do SDN estarem de acordo com a demanda de novas tecnologias como Nuvem (flexibilidade) e *Big Data* (dinamicidade de acesso aos dados).

5.1. Trabalhos Futuros

Nos trabalhos futuros serão adicionados funcionalidades como a possibilidade de incluir características de rede (largura de banda, perda de pacotes, etc.) na etapa de simulação do Mininet. Além de operações que forneçam dados estatísticos sobre a topologia gerada como a média de *hosts* por *switch* e novos modelos probabilísticos.

Referências

- [Calvert et al. 1997] Calvert, K. L., Doar, M. B., and Zegura, E. W. (1997). Modeling internet topology. volume 35, pages 160–163, Atlanta, GA, USA. IEEE.
- [Doar 1996] Doar, M. B. (1996). A Better Model for Generating Test Networks. In *Global Telecommunications Conference, 1996. GLOBECOM'96.'Communications: The Key to Global Prosperity*, pages 86 – 93, Acton, MA, USA.

- [Faloutsos et al. 1999] Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262.
- [Higginbotham 2012] Higginbotham, S. (2012). How google is using openflow to lower its network costs.
- [Hoelzle 2012] Hoelzle, U. (2012). Openflow@ google. *Open Networking Summit*, 17.
- [Kim and Feamster 2013] Kim, H. and Feamster, N. (2013). Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119.
- [Koldehofe et al. 2012] Koldehofe, B., Dürr, F., Tariq, M. A., and Rothermel, K. (2012). The power of software-defined networking: Line-rate content-based routing using openflow. In *Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing*, MW4NG ’12, pages 3:1–3:6, New York, NY, USA. ACM.
- [Lantz et al. 2010] Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 19:1–19:6, New York, NY, USA. ACM.
- [McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- [Medina et al. 2001] Medina, A., Lakhina, A., Matta, I., and Byers, J. (2001). BRITE: An Approach to Universal Topology Generation. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*.
- [Nicol et al. 2003] Nicol, D., Premore, B., Ogielski, A., and Liljenstam, M. (2003). Using simulation to understand dynamic connectivity at the core of the internet. *Proceedings of UKSim*, 2003.
- [Riley 2003] Riley, G. F. (2003). The georgia tech network simulator. In *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, MoMeTools ’03, pages 5–12, New York, NY, USA. ACM.
- [Tangmunarunkit et al. 2002] Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S., and Willinger, W. (2002). Network topology generators: Degree-based vs. structural. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM ’02, pages 147–159, New York, NY, USA. ACM.
- [Waxman 1988] Waxman, B. M. (1988). Routing of Multipoint Connections. In *Selected Areas in Communications, IEEE Journal on*, volume 6, pages 1617–1622, St. Louis, MO, USA. ACM.