

Heurística Construtiva para o Problema de Orquestração da Coleta de Dados de Telemetria *In-band*

Ariel G. Castro¹, Arthur F. Lorenzon¹, Fábio D. Rossi¹, Marcelo C. Luizelli¹

¹Universidade Federal do Pampa (UNIPAMPA)
Alegrete, RS, Brasil

Abstract. *With the increasing usage of networking services, network monitoring has become essential to assure quality of service. In-band Network Telemetry (INT) allows to embed network state information into probe packets. However, INT only defines primitives at device-level, while wide-scale monitoring remains undefined. In this paper, we propose a constructive algorithmic approach to determine the paths taken by probes collecting telemetry items. Results show that our approach generate solutions close to the optimal ones.*

1. Introdução

A proliferação da utilização de serviços em nuvem como, por exemplo, *streaming* de vídeo, exige dos provedores de infraestruturas um cuidado constante com a operação e gerenciamento dos recursos disponíveis e, mais importante, do monitoramento de eventos anômalos (e.g., falhas transientes). Dessa forma, infraestruturas de *data centers* mantêm informações sobre o estado dos recursos existentes na infraestrutura em diferentes níveis de granularidades para garantir seu correto funcionamento [Gupta et al. 2018]. No monitoramento tradicional, protocolos de gerenciamento (e.g., SNMP) são vastamente utilizados para periodicamente requisitar dados de telemetria aos dispositivos de encaminhamento. Entretanto, devido à constante iteração entre o plano de controle e de dados, além das limitações de CPU dos dispositivos, tal mecanismo se mostra ineficaz na coleta das informações sobre o estado da rede – principalmente, aqueles estados relacionados com o plano de dados dos dispositivos de encaminhamento.

A linguagem P4 [Bosshart et al. 2014] surgiu recentemente como uma alternativa para a reprogramação do plano de dados de dispositivos de encaminhamento. P4 permite amenizar o problema de monitoramento de infraestruturas de rede ao criar novos mecanismos para o monitoramento em uma granularidade maior em dispositivos programáveis. Permite-se que pacotes requisitem informações de estado interno dos dispositivos tais como o tamanho da fila, latência de enfileiramento, quando passam pelo *pipeline* do plano de dados, sem intervenção adicional do plano de controle. Dessa forma, é possível injetar *probes* no tráfego de rede. Em cada dispositivo de encaminhamento, ao longo de seu caminho, o *probe* extrai dados de telemetria, até seu último *hop*, onde os dados coletados são enviados a um controlador para análises.

Nesse artigo, propõe-se uma heurística construtiva chamada *Edge Randomization* (ER). A heurística é projetada para construir, de maneira otimizada, ciclos para os *probes* injetados nas infraestruturas física. Com coletores distribuídos em pontos específicos da rede, cria-se iterativamente, ciclos passando por dispositivos vizinhos randomicamente selecionados até que a capacidade do *probe* alcance um certo limiar, retornando ao ponto inicial. Por fim, as soluções propostas apresentaram resultados próximos do ótimo.

O restante do artigo está organizado como segue. Na Seção 2, é apresentado o atual estado da arte. Na Seção 3, é apresentado o problema e a solução proposta. Na Seção 4, os resultados são discutidos. Na Seção 5, conclui-se com as considerações finais.

2. Trabalhos Relacionados

Nesta seção, discute-se os principais trabalhos relacionados com orquestração da coleta de itens de telemetria *in-band*. A telemetria *in-band* surgiu recentemente como uma alternativa para prover maior visibilidade do estado da infraestrutura rede [Ananthanarayanan et al. 2017] em comparação a métodos tradicionais (e.g., SNMP e NetFlow). Dessa forma, é possível detectar/diagnosticar problemas de maneira mais ágil (e.g., *micro-bursts* [Jeyakumar et al. 2014]).

EverFlow [Zhu et al. 2015] e PathDump [Tammana et al. 2016] coletam informações em infraestruturas de rede de *data centers* para auxiliar no *debugging* e gerenciamento da infraestrutura, identificando a origem de falhas que podem causar problemas de roteamento, balanceamento de carga e potenciais *loops*. Ambos os trabalhos utilizam dispositivos de encaminhamento não programáveis para rastrear o caminho dos pacotes através da infraestrutura de rede. EverFlow [Zhu et al. 2015] prioriza a coleta de dados de telemetria a partir dos dispositivos de encaminhamento, enquanto PathDump [Tammana et al. 2016] visa distribuir essa função entre *hosts* e os próprios dispositivos de encaminhamento. Por sua vez, Pan et al. [Pan et al. 2019] propõem a utilização de pacotes *probe* para coletar as informações da infraestrutura. Os caminhos são gerados com base em dispositivos “fontes”, utilizando busca em profundidade e caminhos *eulerianos*.

[de Castro et al. 2019], [Marques et al. 2019], [Hohemberger et al. 2019] deram os primeiros passos em direção à coleta otimizada de dados de telemetria com fluxos ativos (i.e., embutindo informações de telemetria em pacotes de tráfego de rede). Os dois primeiros orquestram como os itens devem ser coletados da infraestrutura através de diferentes heurísticas. Já [Hohemberger et al. 2019], prioriza a coleta dos itens de acordo com as necessidades de aplicações de monitoramento com o auxílio de aprendizagem de máquina. Existem diferentes esforços para orquestrar como as informações de telemetria devem ser coletados na infraestrutura de rede. Porém, pouco ainda foi feito para estabelecer o menor número de caminhos necessários para que diferentes *probes* possam cobrir a rede, diminuindo seu impacto sobre as aplicações existentes.

3. Definição do Problema e Solução Proposta

3.1. Definição do Problema

Neste trabalho, dá-se o primeiro passo em direção à orquestração da coleta de dados utilizando pacotes de *probing*. A Figura 1 ilustra uma infraestrutura de rede composta por um conjunto de dispositivos de encaminhamento (de A até I) e um conjunto de fluxo de probes (f_1 , f_2 , f_3 e f_4), com diferentes itens de telemetria a serem coletados. No exemplo, o fluxo f_1 pode coletar os dados de telemetria dos dispositivos A, C, D, F, G, começando e terminando no dispositivo D. Existe um conjunto de coletores Q posicionados em pontos estratégicos da infraestrutura, onde os *probes* são gerados. Observe que os *probes* gerados são encaminhados através de ciclos previamente determinados de modo a coletarem todas as informações de telemetria da infraestrutura.

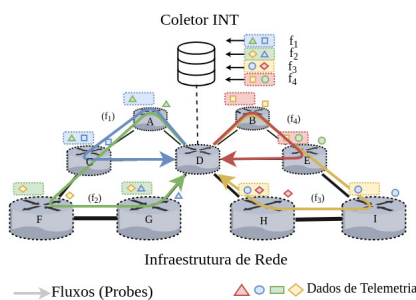


Figure 1. Exemplo de uma solução para o problema utilizando *probes*.

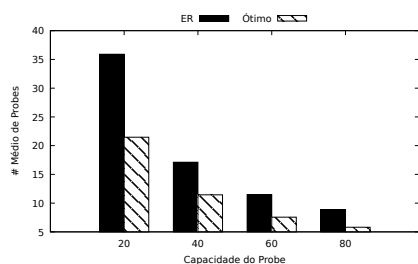
O problema pode ser descrito como um problema de otimização. O objetivo consiste em minimizar o número de *probes* necessários para coletar todas as informações de telemetria e visitar todos os enlaces. Para tanto, considera-se uma infraestrutura física $G = (D, E)$, um conjunto de itens de telemetria V , um conjunto máximo de *probes* P e um conjunto de circuitos de *probes* C . O conjunto D na rede G representa os dispositivos programáveis de encaminhamento $D = \{1, \dots, |D|\}$, enquanto o conjunto E representa os enlaces interconectando pares de dispositivos $(d_1, d_2) \in (D \times D)$. Cada dispositivo $d \in D$ é capaz de encapsular um subconjunto de itens $V_d \in V$ em pacotes *probes* $p \in P$, cada qual com um ciclo único $c \in C$ associado, em um coletor $q \in Q$. O ciclo de um dado *probe* $p \in P$ é mapeado através de uma função $C : P \rightarrow \{D_1 \times \dots \times D_{|D|}\}$. Cada item de telemetria $v \in V$ possui seu tamanho definido por uma função $S : V \rightarrow \mathbb{N}^*$. Os ciclos C são representados por fluxos, os quais são utilizados para a coleta de dados de telemetria V dos dispositivos D . Cada *probe* em seu ciclo possui uma capacidade limitada por uma constante $K_c \in \mathbb{N}^*$. A definição formal completa é omitida devido à limitação de espaço.

3.2. Solução Heurística Proposta

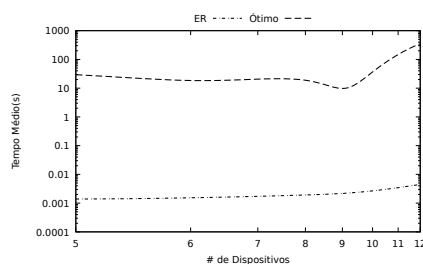
Uma abordagem construtiva (*Edge Randomization – ER*) foi utilizada para geração de ciclos na infraestrutura. O algoritmo iterativamente constrói ciclos através da seleção de enlaces vizinhos ainda não visitados ou cujos dispositivos possuem suas demandas ainda não satisfeitas. Primeiramente, define-se os enlaces não satisfeitos R_E com os mesmos itens do conjunto de arestas existentes E . Seleciona-se aleatoriamente um dispositivo $d \in D$ para servir como ponto inicial e final do ciclo, através da função $RndDev : D \rightarrow D$. Para cada ciclo, enquanto a capacidade utilizada do *probe* não atingir metade de sua utilização, coleta-se os itens de dispositivos d_a e d_b que compõem o enlace $\{d_a, d_b\} \in E$. Por fim, a aresta é satisfeita quando todos os itens de ambos os dispositivos forem coletados, repetindo o processo para os *probes* dos próximos circuitos a serem criados.

4. Avaliação

A seguir, avalia-se a qualidade das soluções geradas. Todos os experimentos foram implementados em linguagem Java e executados em um processador AMD Ryzen Threadripper 2920x com 32GB de RAM. Os resultados obtidos pelo algoritmo proposto foram comparados com os resultados ótimos, implementados através do *solver* IBM CPLEX. Nos experimentos conduzidos, varia-se o tamanho da infraestrutura física entre 4 e 12 dispositivos (conectados de maneira aleatória – i.e., um par de dispositivos está conectado com



(a) Número médio de *probes*



(b) Tempo médio de execução

probabilidade 60%). A capacidade dos *probes* é variada entre 20 e 80 *bytes*. O tamanho dos itens é fixado em 20 *bytes* e o número de itens em cada dispositivo inicia-se em 2 e é incrementado com esse valor até 8 itens.

A Figura 2(a) ilustra o número de ciclos médio utilizados para coletar todas as informações de telemetria da infraestrutura física – variando a capacidade dos *probes* (de 20 a 80 *bytes*) e o número de dispositivos de 4 a 12. No caso médio, a qualidade das soluções da heurística (ER) está 55.4% do caso ótimo. A Figura 2(b) apresenta o tempo médio de execução dos algoritmos para instâncias do problema variando o seu tamanho da mesma forma que na Figura 2(a). Observa-se que o problema, pelo método ótimo, é resolvido em tempo exponencial em relação ao ER.

5. Considerações Finais

Neste artigo, apresentou-se o problema de orquestração da coleta de dados de telemetria *in-band* e foi proposta uma abordagem heurística. Resultados evidenciam que a abordagem proposta gera soluções 55.4% próximas do ótimo em um tempo exponencialmente menor.

References

- Ananthanarayanan, G., Bahl, P., Bodík, P., Chintalapudi, K., Philipose, M., Ravindranath, L., and Sinha, S. (2017). Real-time video analytics: The killer app for edge computing. *computer*, 50(10):58–67.
- Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al. (2014). P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95.
- de Castro, A. G., Vogt, F., Lopes, V. H., da Silva, S. R., Lorenzon, A., Rossi, F., and Luizelli, M. C. (2019). Análise do Desempenho de Heurísticas na Coleta de Informações de Telemetria In-Band. In *ERRC'19*, Alegrete-RS, Brasil.
- Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and Willinger, W. (2018). Sonata: Query-driven streaming network telemetry. In *Proceedings of the SIGCOMM'18*, pages 357–371. ACM.
- Hohemberger, R., Castro, A. G., Vogt, F. G., Mansilha, R. B., Lorenzon, A. F., Rossi, F. D., and Luizelli, M. C. (2019). Orchestrating in-band data plane telemetry with machine learning. *IEEE Communications Letters*.
- Jeyakumar, V., Alizadeh, M., Geng, Y., Kim, C., and Mazières, D. (2014). Millions of little minions: Using packets for low latency network programming and visibility. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 3–14, New York, NY, USA. ACM.
- Marques, J. A., Luizelli, M. C., Tavares da Costa Filho, R. I., and Gaspary, L. P. (2019). An optimization-based approach for efficient network monitoring using in-band network telemetry. *Journal of Internet Services and Applications*, 10(1):12.
- Pan, T., Song, E., Bian, Z., Lin, X., Peng, X., Zhang, J., Huang, T., Liu, B., and Liu, Y. (2019). Int-path: Towards optimal path planning for in-band network-wide telemetry. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 487–495. IEEE.
- Tamma, P., Agarwal, R., and Lee, M. (2016). Simplifying datacenter network debugging with pathdump. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 233–248, Savannah, GA.
- Zhu, Y., Kang, N., Cao, J., Greenberg, A., Lu, G., Mahajan, R., Maltz, D., Yuan, L., Zhang, M., Zhao, B. Y., and Zheng, H. (2015). Packet-level telemetry in large datacenter networks. In *Proceedings of the 2015 ACM Conference on SIGCOMM*, SIGCOMM '15, pages 479–491, New York, NY, USA. ACM.