

CONJUNTO MÍNIMO DE OBSERVADORES PARA GERAÇÃO DE MATRIZES DE TRÁFEGO

Fernando Ávila Fossi Silveira, Renato Elias Nunes de Moraes

Departamento de Computação e Eletrônica (DCEL)

Universidade Federal do Espírito Santo (UFES)

São Mateus/ES

fernandofossi@gmail.com, renato.moraes@ufes.br

Rodolfo da Silva Villaça

Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo (Ifes)

Serra/ES

rodolfo.villaca@ifes.edu.br

RESUMO

A geração de matrizes de tráfego é uma importante função para avaliar a quantidade de fluxos em uma rede. Entretanto, gerar essas matrizes a partir do tráfego medido nos enlaces é uma operação custosa devido à grande quantidade de dados que precisa ser coletada e analisada. Este trabalho busca reduzir o custo de geração dessas matrizes por meio da minimização do número de observadores de cadeias de dados (*data streaming*) instalados nos nós da rede sob avaliação. O problema da instalação de observadores para geração de matrizes de tráfego é modelado como um problema de transversal mínima (*hitting set*). Desta forma, este artigo apresenta um algoritmo (CMO-MT) para reduzir o conjunto de observadores instalados na rede e uma técnica (GeraMatriz) que, a partir do conjunto reduzido de observadores, constrói a matriz de tráfego sem que a sua precisão seja afetada. Extensos experimentos computacionais são apresentados para validar a eficácia da abordagem e do algoritmo proposto.

PALAVRAS CHAVE. Matriz de tráfego, problema da transversal mínima, cadeias de dados.

Tópicos: TEL&SI – PO em Telecomunicações e Sistemas de Informações; TAG – Teoria e Algoritmos em Grafos; OC – Otimização Combinatória.

ABSTRACT

Traffic matrix generation is an important function for evaluating the number of flows in each link in a network. However, the generation of these matrices from measuring the traffic in each link in a network is an expensive operation due to the large amount of data that needs to be collected and analyzed. This paper aims to reduce the cost of generating these matrices by minimizing the number of data streaming observers in the network. The problem of reducing the number of observers installed in the network to generate traffic matrices is modelled as a hitting set problem. An algorithm (CMO-MT) to minimize the number of observers in the network, and a technique (GeraMatriz) to generate the traffic matrix of a network without losing precision are proposed in this paper. Extensive evaluations are presented to validate the effectiveness of the proposed approach and algorithms.

KEYWORDS. Traffic matrix, hitting set problem, data streaming.

Main Area: TEL&SI – PO em Telecomunicações e Sistemas de Informações; TAG – Teoria e Algoritmos em Grafos; OC – Otimização Combinatória.

1. Introdução

Nos últimos anos, a realidade dos usuários das redes de computadores tem sido alterada pela diversidade de conteúdos e serviços oferecidos tanto para usuários domésticos como para estabelecimentos comerciais e instituições públicas. A maioria das aplicações disponíveis exigem alto índice de Qualidade de Serviço (QoS) pois a perda e o atraso de pacotes tem grande impacto em sua usabilidade. Aliado ao cenário de expansão da Internet, tem-se o aumento da popularidade e capacidade de processamento de dispositivos e conexões móveis que, segundo Sandvine [2014], serão responsáveis pela geração da maior parte do tráfego de dados no futuro. Essas tendências tornam cada vez mais importante a necessidade de se conhecer e analisar o tráfego de dados das redes de computadores.

O conhecimento do tráfego nas redes de computadores é de vital importância para sua operação contínua. A coleta e análise das características do tráfego de dados permite aos operadores prover capacidade de rede suficiente para o tráfego atual, bem como se antecipar e se preparar para o futuro próximo. Dentre as técnicas existentes para a caracterização do tráfego em redes de computadores encontra-se a matriz de tráfego [Paul Tune, 2013]. Uma matriz de tráfego é uma representação abstrata do volume de dados trafegados entre conjuntos de pares de nós em uma rede. Cada elemento da matriz representa o volume de tráfego entre um par de nós da rede em avaliação. O volume de dados geralmente é medido em número de *Bytes* ou quantidade de pacotes.

Os desafios na construção de uma matriz de tráfego estão associados à necessidade de se conhecer exatamente o volume de dados que trafegou por cada nó ou enlace da rede. Geralmente, observadores de tráfego são instalados nos elementos de rede (roteadores) para permitir uma coleta minuciosa de informações sobre os pacotes. Entretanto, os roteadores usados na implementação dos nós da rede possuem fortes restrições na capacidade de processamento e de memória. Além disso, a grande quantidade de informações coletadas têm que ser enviada, posteriormente, para um local centralizado para análise: a estação de monitoramento de rede. Esta abordagem gera uma quantidade significativa de tráfego que compartilha a mesma infra-estrutura de rede com serviços de usuário. Do ponto de vista desses serviços, esse tráfego é uma sobrecarga, uma vez que ele interfere na disponibilidade de largura de banda para os dados. Ou seja, a própria atividade de monitoramento interfere no tráfego de rede.

Assim, para a diminuição do custo de construção de uma matriz de tráfego, este trabalho propõe a combinação de duas abordagens utilizadas na literatura. Primeiro, utilizar uma técnica de geração de matrizes de tráfego que substitua a coleta minuciosa de informações pela utilização de algoritmos para cadeias de dados (*data streaming algorithms*) [Zhao et al., 2005]. Algoritmos observadores para cadeias de dados permitem o processamento de longas cadeias de dados em uma única passagem, usando uma pequena memória de trabalho (*sketch*) instalada no roteador. Segundo, reduzir a quantidade de observadores a serem implantados [Suh et al., 2006] na rede sem que a construção da matriz de tráfego pela técnica proposta seja afetada. Apesar dos algoritmos observadores para cadeias de dados poderem ser implementados com baixa complexidade de armazenamento e processamento [Zhao et al., 2005], sua instalação em muitos equipamentos de rede, além da sobrecarga de tráfego já discutida, incorre também em custos fixos de implantação tais como custo de hardware/software e custo de manutenção, pois nem todos os dispositivos de rede já instalados estão aptos a armazenar e instanciar os *sketches*. Portanto, para a geração das matrizes de tráfego em uma rede real é importante que se minimize a quantidade de observadores instalados na rede.

A partir dessas características e restrições, neste trabalho são propostos uma técnica de geração de matrizes de tráfego e o problema de escolha do conjunto mínimo de nós observadores para geração de matrizes de tráfego que consiste em: dado o conjunto de nós da rede e o conjunto de caminhos mais curtos entre todos os pares de nós, definir a quantidade mínima de nós observadores e suas posições de maneira que todos os caminhos do conjunto de caminhos mais curtos contenham, pelo menos, um nó observador, e de maneira que a precisão na construção da matriz de tráfego pela

técnica de geração de matrizes proposta não seja afetada. Esse problema é modelado como um problema de transversal mínima (*hitting set*), e os resultados apresentados neste artigo permitem afirmar que é possível implantar nós observadores em apenas uma fração dos nós da rede sem perda de precisão na matriz de tráfego gerada.

O restante deste artigo está organizado na seguinte forma: a Seção 2 faz um levantamento dos trabalhos relacionados na literatura. A Seção 3 descreve o modelo da rede e formaliza a definição do problema do conjunto mínimo de observadores para geração de matrizes de tráfego como um problema de transversal mínima (*hitting set*). A Seção 4 apresenta os algoritmos e a técnica de geração de matrizes de tráfego propostos. Extensos resultados computacionais são apresentados na Seção 5 enquanto na Seção 6 conclui-se o artigo e faz-se o apontamento de trabalhos futuros e em andamento.

2. Trabalhos Relacionados

A abordagem apresentada nesse artigo relaciona o problema da estimativa de matriz de tráfego [Medina et al., 2002; Zhao et al., 2005; Zhang et al., 2009; Wu et al., 2012] com o problema de localização de observadores de tráfego na rede [Cantieni et al., 2006; Suh et al., 2006; Breitbart et al., 2009].

A fim de observar uma grande fração do tráfego de uma rede, é necessário monitorar vários enlaces ao mesmo tempo, uma vez que apenas uma parte do tráfego pode ser vista por um único ponto de medição de uma grande rede. O posicionamento de um observador incorre em custos de implantação, hardware/software, armazenamento e manutenção. Assim, o problema de localização de observadores consiste em selecionar o menor número de posições onde instalar os observadores na rede.

Identificar os locais estratégicos para os observadores de tráfego é um problema difícil que tem atraído interesse significativo na literatura [Cantieni et al., 2006]. Várias soluções têm sido propostas para diferentes contextos. Por exemplo, em [Jamin et al., 2000], os autores propõem a colocação de dispositivos de medição para a construção de mapas de distância. Assume-se que quanto maior a distância entre dois nós, maior será a latência de rede, daí a necessidade desse mapa. Outros trabalhos têm abordado o problema de posicionamento dos observadores para monitoramento ativo (aquele cuja medição interfere no tráfego da rede) da infraestrutura para medir atrasos e detectar falhas de enlace [Horton e López-Ortiz, 2003; Nguyen e Thiran, 2004; Bejerano e Rastogi, 2006].

Considerando monitoramento passivo (aquele cuja medição não interfere no tráfego da rede), Suh et al. [2006] tratam o problema de posicionar os observadores e definir taxas de amostragem de pacotes (*sampling rate*) a fim de maximizar a fração de fluxos a ser amostrado em função dos custos de operação e instalação. Cantieni et al. [2006] também determinam o conjunto mínimo de observadores e suas taxas de amostragem a fim de obter uma medição de alta precisão com baixo consumo de recursos.

Neste trabalho, o conjunto mínimo de nós observadores e suas posições deve ser determinado de maneira que a construção da matriz de tráfego pela técnica de geração de matrizes de tráfego com utilização de cadeia de dados (*data streaming*) proposta possa ser utilizada como alternativa à técnica de amostragem sem que haja perda na precisão da matriz de tráfego gerada. Na técnica a ser proposta, um algoritmo de cadeia de dados de rede irá processar os pacotes que passam por um enlace a fim de recolher a informação mais importante, que permitirá responder a um tipo específico de consulta, usando uma pequena estrutura de dados do tipo *counter array* [Flajolet e Martin, 1985]. No caso deste artigo, a "pergunta" que se deseja responder é: "como está o volume de tráfego entre todos os pares de nós de uma rede?". Essa pergunta pode ser respondida por meio da geração da matriz de tráfego da rede. *Network data streaming* tem sido reconhecida como a melhor alternativa para medição e monitoramento de enlaces de redes de alta velocidade em comparação à amostragem de pacotes [Zhao et al., 2005].

Desta forma, o problema a ser resolvido consiste em selecionar o menor número de nós observadores e suas posições (problema de posicionamento de observadores) de maneira que a precisão da matriz de tráfego estimada usando os algoritmos de cadeia de dados instalados nos nós observadores não seja prejudicada (problema da estimativa de matriz de tráfego de uma rede).

3. Modelo de Rede e Formulação do Problema

Uma rede pode ser definida como um grafo conexo $G = (V, A)$ onde os roteadores são representados pelo conjunto de nós V numerados de 0 a $|V| - 1$. O conjunto de arestas A contém pares ordenados de nós (u, v) , com $u, v \in V$, representando o enlace direcionado de u para v . Quando a aresta (u, v) está presente em A , significa que informações podem ser enviadas diretamente de u para v através do enlace (u, v) . Nas redes consideradas, as conexões entre os roteadores são sempre bidirecionais, ou seja, dado $u, v \in V$, informações podem ser enviadas de u para v e vice-versa. Assim, sempre que (u, v) estiver presente em A , também estará presente em A o enlace (v, u) .

A unidade básica de informação trafegando na rede é chamada de pacote. Todo pacote está associado a um par de nós $[v_i, v_e]$, representado como $p_{[v_i, v_e]}$, tal que $v_i \in V$ é o nó origem (ingresso) e $v_e \in V$ é o nó destino (egresso) do pacote. Um pacote então viajará pela rede partindo do seu nó ingresso v_i até o seu nó egresso v_e passando por um conjunto de enlaces $E_{[v_i, v_e]} \subset A$. Esse conjunto de enlaces é chamado de caminho. Como um enlace é representado pelo par de nós (u, v) pode-se representar também um caminho pelo conjunto de nós que formam os enlaces da seguinte forma: $C_{[v_i, v_e]} = \{u \in V \mid (u, v) \in E_{[v_i, v_e]} \text{ ou } (v, u) \in E_{[v_i, v_e]}\}$. Como exemplo, a Figura 1 mostra uma topologia com os nós $V = \{0, 1, \dots, 7\}$ conectados pelos enlaces exibidos como linhas entre eles. Um pacote $p_{[0,7]}$ passa pelo conjunto de enlaces $E_{0,7} = \{(0, 3), (3, 4), (4, 7)\}$, conseqüentemente, pelo conjunto de nós $C_{[0,7]} = \{0, 3, 4, 7\}$. Podem existir roteadores $v \in V$ que não são ingresso nem egresso de nenhum pacote, mas podem participar de seus caminhos. Na Figura 1, os nós 3 e 4 podem não ser ingresso nem egresso de nenhum pacote, mas participam de todos os caminhos da rede.

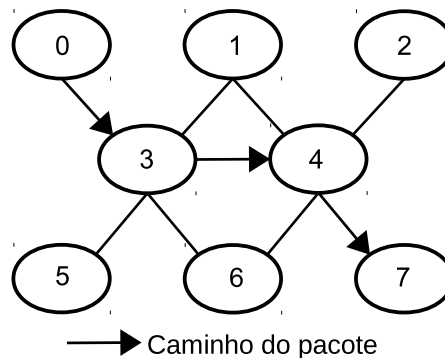


Figura 1: Exemplo de um pacote trafegando em uma topologia de rede.

Entre um par de nós ingresso-egresso $[v_i, v_e]$ podem existir vários caminhos. Geralmente, esses caminhos podem ser inferidos a partir tabela de roteamento do roteador e dependem dos protocolos de roteamento utilizados pela rede. Nesse trabalho é considerado o menor caminho entre os dois nós pois, do ponto vista de protocolos de roteamento bastante conhecidos, tais como OSPF (*Open Shortest-Path First*) e IS-IS (*Intermediate System to Intermediate System*), pacotes com origem em v_i e destino em v_e trafegam pelo menor caminho de v_i para v_e [Wu et al., 2012]. Como não há peso nas arestas do grafo $G = (V, A)$, o caminho $E_{[v_i, v_e]}$ é definido como o conjunto com o menor número de arestas que conecta v_i a v_e . O menor caminho entre um par de nós ingresso-egresso $[v_i, v_e]$ pode ser calculado pelo Algoritmo de Dijkstra [Cormen et al., 2009].

O problema do conjunto mínimo de observadores para geração de matrizes de tráfego (CMO-MT) consiste em, dado o grafo $G = (V, A)$, o conjunto de pacotes P que devem trafegar pela rede e o conjunto de caminhos $\mathcal{C} = \{C_1, C_2, \dots, C_j\}$ tal que j é a quantidade de pares de nós ingresso-egresso na rede e $j \leq |V|(|V| - 1)$, definir o conjunto mínimo de observadores $O \subseteq V$ tal que a precisão na construção da matriz de tráfego a ser gerada pela técnica discutida na Seção 4 não seja afetada.

Para a construção da matriz de tráfego é necessário monitorar o tráfego da rede. Esse monitoramento consiste em contar quantos pacotes trafegaram entre todos os pares de nós ingresso-egresso $[v_i, v_e]$. Quando um pacote $p_{[v_i, v_e]}$ viaja de v_i para v_e , ele passa por todos os nós do caminho $C_{[v_i, v_e]}$. Portanto, para contar quantos pacotes trafegaram entre um determinado par de nós ingresso-egresso $[v_i, v_e]$, pode-se instalar um único observador em um nó $u \in C_{[v_i, v_e]}$. Para monitorar toda a rede, deve-se instalar observadores em um conjunto de nós $O \subseteq V$, tal que o conjunto O cubra todos os caminhos $C_{[v_i, v_e]}$ da rede. Como exemplo, na Figura 1, o conjunto mínimo de observadores é $O = \{3, 4\}$, pois estes nós cobrem todos os caminhos presentes na rede, logo todos os pacotes passam por eles.

O problema CMO-MT, que consiste em encontrar o conjunto de observadores O de menor tamanho $|O|$ que cubra todos os caminhos $C_{[v_i, v_e]}$ da rede, pode ser modelado como um problema de *Hitting Set*, ou problema de transversal mínima. O problema de transversal mínima é um importante membro da classe NP-Difícil [Garey e Johnson, 1979; Caprara et al., 1999] e pode ser definido como: dados um conjunto finito F , uma coleção finita $\mathcal{S} = \{S_1, \dots, S_j\}$ de subconjuntos de F , encontrar um conjunto $B \subseteq F$ de cardinalidade $|B|$ mínima, tal que $B \cap S_i \neq \emptyset$ para todo $i = 1, \dots, j$.

O problema CMO-MT pode então ser convertido em um problema de transversal mínima da seguinte forma: dados o conjunto V de nós, o conjunto $\mathcal{C} = \{C_1, C_2, \dots, C_j\}$, $j \leq |V|(|V| - 1)$, de caminhos entre os pares de nós (subconjuntos de V), encontrar o conjunto de observadores $O \subseteq V$ de cardinalidade $|O|$ mínima, tal que $O \cap C_i \neq \emptyset$ para todo $i = 1, \dots, j$.

4. Algoritmo para Solução do Problema CMO-MT

O algoritmo de solução para o problema CMO-MT está inserido em uma técnica proposta nesse artigo, intitulada GeraMatriz, que divide o sistema de geração de matrizes de tráfego em três fases. A Figura 2 mostra a arquitetura do sistema. Na primeira fase, a Estação de Monitoramento recolhe todas as informações necessárias e executa o algoritmo para solução do problema CMO-MT, cujo resultado é o conjunto de nós que deve receber o algoritmo observador. Na segunda fase, os nós escolhidos executam os algoritmos observadores para contagem dos pacotes. Na terceira fase, a Estação de Monitoramento recebe as informações de contagem enviadas pelos nós observadores (nós onde o algoritmo observador foi executado) e constrói a matriz de tráfego. A seguir, cada algoritmo é descrito de forma mais detalhada.

O Algoritmo 1 descreve o algoritmo para solução do problema CMO-MT executado na fase 1. Por se tratar de um problema NP-Difícil [Garey e Johnson, 1979; Caprara et al., 1999], foi implementada uma heurística construtiva baseada no algoritmo proposto por Chvatal [1979]. O Algoritmo 1 recebe como entrada o Grafo $G = (V, A)$ e o conjunto de caminhos \mathcal{C} . Na linha 2 o conjunto de observadores O é inicializado como um conjunto vazio. O laço da linha 3 a 7 executa enquanto o conjunto de caminhos não é vazio. Na linha 4, a função `EscolheObservador()` recebe o conjunto $V \setminus O$ de nós que ainda não são observadores e o conjunto \mathcal{C} de caminhos ainda não cobertos por pelo menos um nó observador. A função `EscolheObservador()` seleciona do conjunto $V \setminus O$ o nó contido no maior número de caminhos de \mathcal{C} e o retorna como o novo nó a receber o algoritmo observador. Em seguida, o conjunto \mathcal{C} de caminhos ainda não cobertos é atualizado pela remoção dos caminhos que contém o novo observador obs na linha 5 e o conjunto O de observadores é atualizado com a inclusão de obs na linha 6. O conjunto final de nós de observadores é retornado na linha 8.

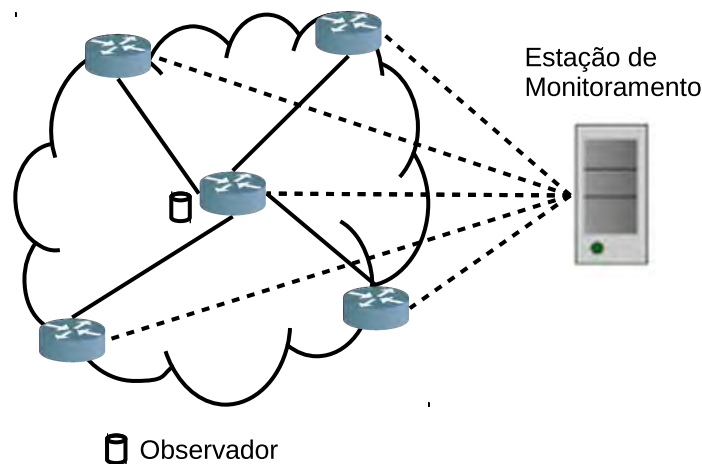


Figura 2: Arquitetura do sistema de geração de matrizes com alocação mínima de observadores.

O Algoritmo 1 foi implementado com complexidade $O(|V||\mathcal{C}| \min(|V|, |\mathcal{C}|))$ [Cormen et al., 2009]. Chvatal [1979] demonstrou que esse algoritmo alcança taxa de aproximação da ordem de $\ln |\mathcal{C}| + 1$.

Algoritmo 1: Solução do problema CMO-MT

Entrada: Grafo $G = (V, A)$, conjunto de caminhos \mathcal{C}

Saída: Conjunto de nós de observadores O

```

1 início
2    $O \leftarrow \emptyset$ 
3   enquanto  $\mathcal{C} \neq \emptyset$  faça
4      $obs \leftarrow \text{EscolheObservador}(V \setminus O, \mathcal{C})$ 
5      $\mathcal{C} \leftarrow \text{RemoveCaminhos}(\mathcal{C}, obs)$ 
6      $O \leftarrow O \cup \{obs\}$ 
7   fim enqto
8   retorna  $O$ 
9 fim
    
```

Durante a fase 2, os nós observadores selecionados pelo Algoritmo 1 executarão o algoritmo observador para realizar a contagem dos pacotes que trafegam pelos caminhos da rede. O algoritmo observador é um algoritmo para cadeia de dados que utiliza uma estrutura de dados do tipo *counter-array* [Flajolet e Martin, 1985]. O *counter-array* implementado é uma matriz M , onde cada posição $M_{[v_i, v_e]}$ irá armazenar a quantidade de pacotes trafegados no caminho $C_{[v_i, v_e]}$. O tamanho da matriz pode ser de até $|V| \times |V|$, visto que, no pior caso, todos os nós participam como ingresso e egresso. Inicialmente, a matriz é inicializada com zeros em todas as suas posições. Toda vez que um pacote $p_{[v_i, v_e]}$ passa por um nó observador u no caminho $C_{[v_i, v_e]}$, o elemento $M_{[v_i, v_e]}$ da matriz armazenada na memória de u é incrementado e o pacote encaminhado.

Na fase 3, a matriz de tráfego (*MT*) da rede é gerada periodicamente (por exemplo, de 10 em 10 minutos) pela Estação de Monitoramento. Ao final do período de contagem, em cada nó observador, o *counter-array* é movido da memória para o disco e um novo *counter-array* com as posições zeradas é criado na memória. A Estação de Monitoramento, além de gerenciar o período de contagem, também é responsável por recolher o *counter-array* no disco de cada nó observador.

Recolhidos os *counter-arrays* de todos os nós observadores, a Estação de Monitoramento constrói a matriz *MT* com tamanho de até $|V| \times |V|$, no pior caso. A matriz de tráfego *MT* é gerada pela estação da seguinte maneira: inicializa *MT* com todos os elementos zerados, para cada caminho $C_{[v_i, v_e]} \in \mathcal{C}$, recupera a contagem dos pacotes do *counter-array* do nó observador

que cobre o caminho $C_{[v_i, v_e]}$ e atualiza a posição $MT_{[v_i, v_e]}$. No final do procedimento de geração tem-se a matriz de tráfego MT onde cada posição $MT_{[v_i, v_e]}$ representa a quantidade de pacotes trafegados no caminho entre o par de nós ingresso-egresso $[v_i, v_e]$.

5. Avaliações e Resultados

Para avaliar a solução para o problema CMO-MT e a técnica de geração de matrizes de tráfego GeraMatriz, ambas propostas neste artigo, foram utilizados *traces* anonimizados de dados de tráfego real¹ e topologias reais de Sistemas Autônomos (ASes) da Internet², obtidos diretamente do CAIDA (*Center for Applied Internet Data Analysis*).

Foram utilizados 8 diferentes grafos nos testes computacionais. Cada grafo representa a topologia de rede de um sistema autônomo (AS) diferente com quantidade de nós $|V|$ variando de 11 a 115. Nos testes realizados, todos os nós participam como ingresso e egresso, configurando o pior caso onde tem-se $|V|(|V| - 1)$ pares ingresso-egresso. Os *counter-arrays* dos nós observadores e a matriz de tráfego MT gerada têm dimensões $|V| \times |V|$. Para cada grafo, foram colocados em circulação um total de 100.000 pacotes retirados dos *traces*. Os mesmos 100.000 pacotes foram utilizados em todos os testes de tal forma a caracterizar o mesmo tráfego em todos os ASes.

No primeiro experimento computacional, o objetivo é avaliar a redução do número de nós observadores computada pelo Algoritmo 1. A Tabela 1 apresenta, para todos os 8 sistemas autônomos avaliados, a saber: 1) Abilene; 2) AS-1221; 3) AS-1239; 4) AS-2914; 5) AS-3257; 6) AS-3356; 7) AS-7018 e 8) Geant, a quantidade de nós observadores necessários para geração das matrizes de tráfego dada pelo Algoritmo 1 e o percentual de redução obtido em relação ao total de nós da rede, já que, nos experimentos, todos os nós participam como ingresso e egresso e, assim, todos os nós são potenciais candidatas a receberem o algoritmo observador.

Tabela 1: Redução do número de observadores nos 8 ASes avaliados.

AS	$ V $	Nós observadores	Redução
Abilene	11	7	36%
AS-1221	44	6	86%
AS-1239	52	22	58%
AS-2914	70	29	59%
AS-3257	41	16	61%
AS-3356	63	20	69%
AS-7018	115	21	82%
Geant	22	12	45%

Conforme pode ser observado na Tabela 1 o algoritmo proposto conseguiu reduzir o número de observadores em todos os casos, variando entre 36% e 86%. Como exemplo, a Figura 3 apresenta a topologia do AS-3356 ($|V| = 41$), com os 16 nós observadores destacados em vermelho.

Após a instalação dos observadores nos nós indicados pelo algoritmo de redução, procedeu-se à simulação do tráfego de 100.000 pacotes. Em cada nó observador executou-se o algoritmo para cadeia de dados com memória *counter-array* da fase 2 da técnica GeraMatriz. Após a fase 2, a matriz de tráfego foi gerada na fase 3 como descrito na Seção 4.

A precisão da matriz de tráfego gerada pela técnica GeraMatriz pode ser estabelecida através da sua comparação com a matriz de tráfego real (contagem exaustiva da quantidade total de pacotes que trafegaram entre todos os pares ingresso-egresso). Apesar da matriz de tráfego real

¹http://www.caida.org/data/passive/passive_2012_dataset.xml

²<https://www.caida.org/research/topology/#Datasets>

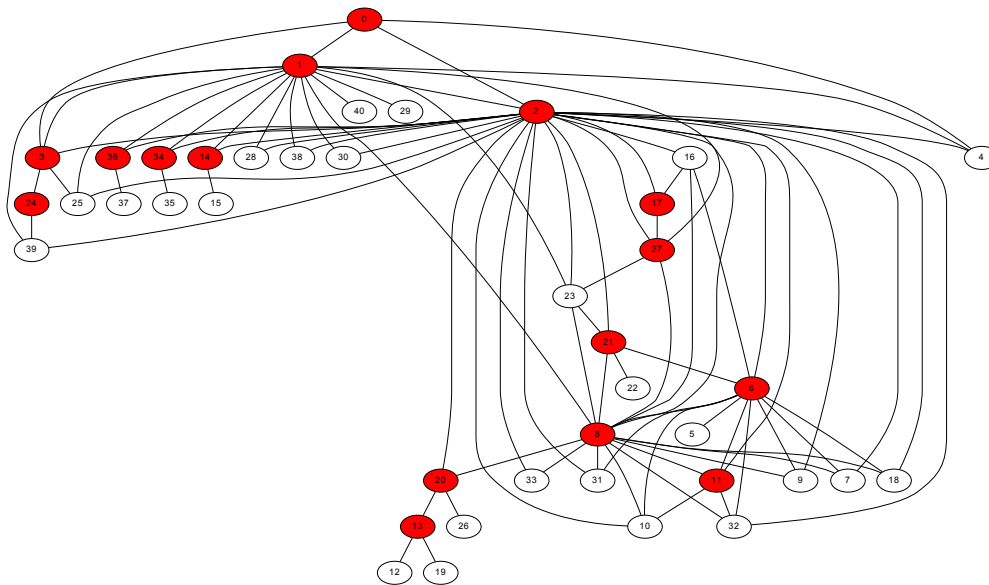


Figura 3: Topologia do AS-3356, com destaque em vermelho para o posicionamento dos 16 nós observadores.

poder ser gerada em ambiente de simulação, em uma rede operacional a geração da matriz real é impraticável devido ao conflito entre o volume de pacotes circulantes, as restrições de memória dos roteadores e a enorme quantidade de tráfego que seria encaminhado para contagem e geração da matriz na estação de monitoramento da rede. Na Técnica GeraMatriz a memória utilizada está restrita aos *counter arrays*, uma estrutura de dados compacta (alguns *kB*) conforme apresentado em Flajolet e Martin [1985].

A Figura 4 mostra a comparação, em números de pacotes, entre a matriz de tráfego gerada pela técnica GeraMatriz e a matriz real calculada na simulação. Para apresentação da comparação foi escolhida a topologia do AS-3356 com $|V| = 41$ (ver Figura 3). O AS-3356 foi escolhido pois seu comportamento representa o comportamento padrão em todos os testes e seu tamanho é mais adequado a uma melhor visualização no espaço deste artigo.

Na Figura 4, as linhas tracejadas representam um erro de $\pm 20\%$ no quantitativo de tráfego da matriz gerada, e foram plotados para servirem de referência do nível de erro obtido com a técnica GeraMatriz. Os pontos pretos representam o erro na estimativa do volume de tráfego da matriz gerada pela técnica GeraMatriz em relação à matriz de tráfego real. Como os pontos pretos estão exatamente sobre a linha de referência, pode-se concluir que a técnica GeraMatriz proposta neste artigo, mesmo utilizando uma pequena memória de trabalho do tipo *counter arrays* e um número reduzido de nós observadores, reproduz a matriz de tráfego real sem erros.

Como em ambientes de redes operacionais a matriz de tráfego real não pode ser calculada, técnicas para estimar a matriz de tráfego são utilizadas. No segundo experimento, a matriz de tráfego gerada pela solução proposta é comparada à matriz de tráfego estimada pela técnica proposta por Zhao et al. [2005]. A técnica de Zhao et al. [2005] também utiliza um algoritmo para cadeias de dados e uma pequena memória de trabalho chamada de *bitmap*. A principal diferença entre a técnica GeraMatriz e a técnica de [Zhao et al., 2005] está na forma como o tamanho da memória de trabalho é gerenciada. Na técnica GeraMatriz o tamanho do *counter array* é dependente de $|V|$, ou seja, em redes com grande quantidade de nós a memória dos roteadores se torna um gargalo. Na técnica de [Zhao et al., 2005] o tamanho do *bitmap* é um parâmetro definido pelo programador, ou seja, que pode ser ajustado de acordo com a memória disponível. Em compensação, o tamanho do *bitmap* está diretamente relacionado com a precisão da estimativa da matriz de tráfego. Como visto

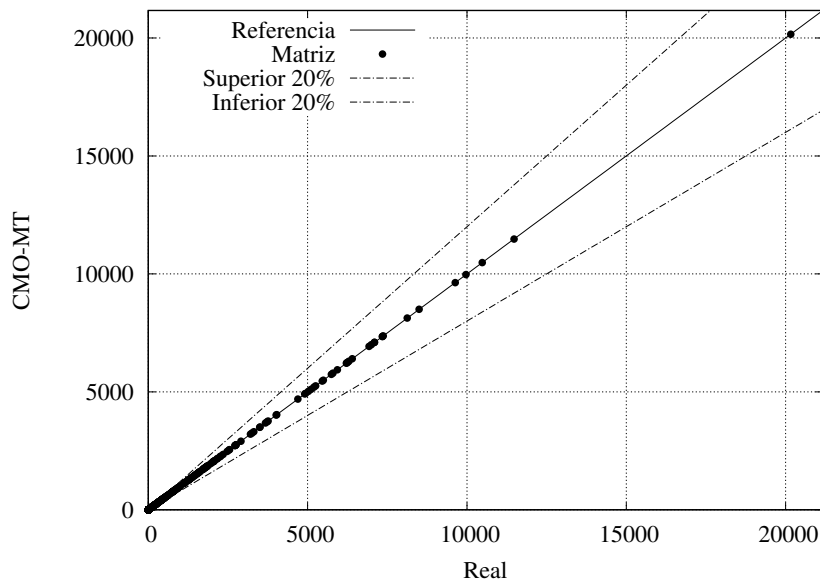


Figura 4: Comparação entre as matrizes gerada pela quantidade reduzida de observadores e a matriz real.

nos resultados do primeiro experimento, não há perda de precisão na geração da matriz de tráfego pela técnica GeraMatriz.

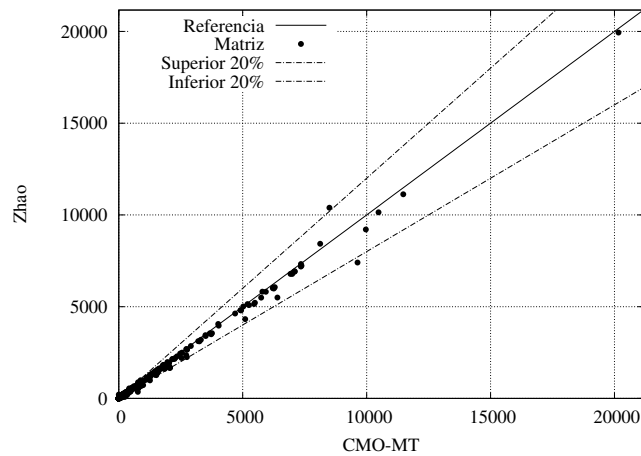
Para facilitar a compreensão dos resultados, é importante ressaltar o compromisso existente na configuração dos *bitmaps* na técnica apresentada por Zhao et al. [2005]: a precisão da matriz gerada é diretamente proporcional ao tamanho dos *bitmaps*. Devido às fortes restrições quanto ao tamanho de memória disponível nos equipamentos de rede, algumas dezenas de *kB*, os *bitmaps* tendem a ser bem menores que a quantidade de caminhos a serem observados, sendo necessária assim a utilização de funções de *hashing* para contagem dos pacotes. Entretanto, quanto menor forem os *bitmaps*, maiores são as chances de ocorrerem colisões nas funções de *hashing* utilizadas na marcação dos pacotes que passam pelo equipamento. Como solução de compromisso, os *bitmaps* são instanciados com um tamanho adequado à realidade dos roteadores e são renovados após um determinado número de acessos, proporcional ao seu tamanho. Quanto menor for esse percentual de acessos, menores serão as chances de ocorrerem colisões nas marcações dos pacotes e, conseqüentemente, maior será a precisão da matriz gerada. Ao mesmo tempo, quanto menores forem os percentuais de ocupação, mais *bitmaps* precisarão ser transportados até a estação de monitoramento para serem contados e, conseqüentemente, gerarem a matriz de tráfego da rede.

Como as técnicas se diferenciam na forma como a memória de trabalho é definida, para uma avaliação justa, certificou-se de que o mesmo espaço de memória foi utilizado por ambas as técnicas: *bitmap* de Zhao et al. [2005] e *counter-array* da GeraMatriz. Essa condição é importante pois, conforme já discutido, o tamanho da memória alocado para os observadores é um ponto crítico dos roteadores.

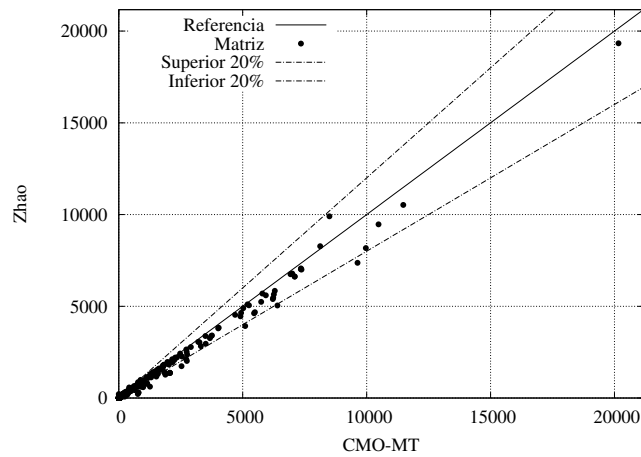
Para apresentação da comparação entre as técnicas foi mantida a topologia do AS-3356 com $|V| = 41$ (ver Figura 3). Foram avaliados cenários com limites de acessos iguais a 10%, 30% e 50% do tamanho total dos *bitmaps*, igual a 6,57 *kB* neste AS.

Na técnica proposta em Zhao et al. [2005] é necessária a instalação de observadores com *bitmaps* em todos os 41 nós para geração da matriz de tráfego do AS-3356. Como observado na Figura 3 com a técnica proposta neste artigo, dos 41 possíveis nós candidatos a observadores, foi necessária a instalação de observadores em apenas 16 deles.

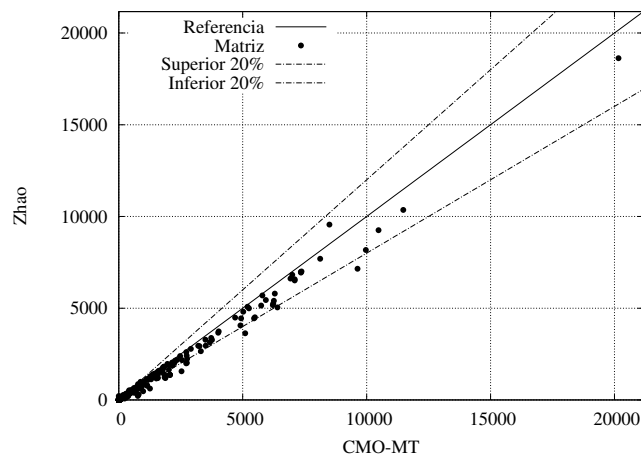
As Figuras 5(a), 5(b) e 5(c) mostram a comparação, em números de pacotes trafegados em cada par de nós (ou seja, em cada posição da matriz), entre a matriz de tráfego gerada pela técnica



(a) *Bitmaps* - Limite 10%.



(b) *Bitmaps* - Limite 30%.



(c) *Bitmaps* - Limite 50%.

Figura 5: Comparação entre as matrizes estimadas e a matriz real.

de Zhao et al. [2005] e a matriz de tráfego gerada pela técnica GeraMatriz. As linhas tracejadas representam $\pm 20\%$ no quantitativo de tráfego da matriz gerada pela técnica GeraMatriz e foram

plotadas para servirem de referência do nível de erro obtido com a técnica de Zhao et al. [2005]. Os pontos pretos representam o erro na estimativa do volume de tráfego da matriz gerada pela técnica de Zhao et al. [2005] em relação à matriz gerada pela técnica GeraMatriz. As Figuras 5(a), 5(b) e 5(c) apresentam os resultados obtidos para 3 (três) diferentes configurações de *bitmaps*, respectivamente, com limites de acessos iguais a 10%, 30% e 50% do tamanho total dos *bitmaps*, fixado em 6,57 *kB* neste AS.

A partir dos resultados apresentados nas Figuras 5(a), 5(b) e 5(c) é possível verificar que a técnica de Zhao et al. [2005] produz matriz de tráfego estimada próxima da matriz gerada pela técnica GeraMatriz, porém com observadores em todos os nós. A técnica de Zhao et al. [2005] é mais precisa quanto menor for o limite de acessos (redução de colisões de *hashing*). Todas essas conclusões a respeito do AS-3356 se repetiram nas outras 7 topologias de rede simuladas.

A partir dos resultados apresentados nesse trabalho, pode-se confirmar que a técnica proposta reproduz a matriz de tráfego real sem erros utilizando-se de um menor número de nós observadores e, em cada observador, as estruturas de dados compactas (*sketches*) do tipo *counter arrays* ocupam exatamente o mesmo espaço em memória nos roteadores que os *bitmaps* propostos na técnica de Zhao et al. [2005].

6. Conclusões e Trabalhos Futuros

A principal contribuição deste artigo foi mostrar que é possível reduzir a quantidade de nós observadores necessários para geração de matrizes de tráfego modelando-o como um problema da transversal mínima (*hitting set*) e usando algoritmos de cadeias de dados com estruturas de dados compactas do tipo *counter-arrays*. Nesse contexto, foi apresentado um algoritmo de solução para o problema do conjunto mínimo de observadores para geração de matrizes de tráfego e uma técnica para a geração da matriz de tráfego a partir dos dados coletados pelos nós observadores por meio desses *counter-arrays*.

A proposta foi avaliada usando *traces* reais de pacotes e topologias de sistemas autônomos da Internet, obtidos no CAIDA. Os resultados mostraram que, se forem obedecidos os relaxamentos impostos pelo modelo proposto (por exemplo, tamanho da memória de trabalho em função do número de nós da rede), a abordagem apresentada consegue gerar matrizes de tráfego sem nenhuma perda de precisão.

Como proposta de trabalhos futuros pretende-se impor algumas restrições ao modelo e avaliar o impacto destas restrições na precisão da matriz gerada. Dentre essas restrições estão: limitar o tamanho da memória e, em consequência, permitir e tratar colisões de *hashing* nos contadores por meio de instalação de observadores em caminhos disjuntos e; não assumir conhecimento prévio, por parte dos elementos de rede, sobre o mapeamento de cada pacote aos seus respectivos nós ingresso-egresso (implementar esse mapeamento de forma distribuída). Além disso, pretende-se estudar a viabilidade de redução de nós observadores mantendo-se a técnica original proposta por Zhao et al. [2005], ou seja, usando *bitmaps*.

Agradecimentos

Esse trabalho foi parcialmente suportado pelo CNPq (processos 461286/2014-9 e 449369/2014-5) e FAPES (processo 524/2015).

Referências

- Bejerano, Y. e Rastogi, R. (2006). Robust monitoring of link delays and faults in ip networks. *IEEE/ACM Transactions on Networking (TON)*, 14:1092–1103.
- Breitbart, Y., Dragan, F. F., e Gobjuka, H. (2009). Effective monitor placement in internet networks. *Journal of Networks*, 4:657–666.
- Cantieni, G. R., Iannaccone, G., Barakat, C., Diot, C., e Thiran, P. (2006). Reformulating the monitor placement problem: Optimal network-wide sampling. In *Proceedings of the 2006 ACM CoNEXT Conference*, p. 5:1–5:12.

- Caprara, A., Fischetti, M., e Toth, P. (1999). A heuristic method for the set covering problem. *Operations Research*, 47(5):730–743. URL <http://pubsonline.informs.org/doi/abs/10.1287/opre.47.5.730>.
- Chvatal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4:233–235.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C. (2009). *Introduction to Algorithms*. MIT Press, Massachusetts/USA, 3 edition. ISBN 978-0262033848.
- Flajolet, P. e Martin, G. N. (1985). Probabilistic Counting Algorithms for Data Base Applications. *J. Comput. Syst. Sci.*, 31(2):182–209. ISSN 0022-0000.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco.
- Horton, J. D. e López-Ortiz, A. (2003). On the number of distributed measurement points for network tomography. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, p. 204–209. ACM.
- Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., e Zhang, L. (2000). On the placement of internet instrumentation. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, p. 295–304. IEEE.
- Medina, A., Taft, N., Salamatian, K., Bhattacharyya, S., e Diot, C. (2002). Traffic matrix estimation: Existing techniques and new directions. *SIGCOMM Comput. Commun. Rev.*, 32:161–174.
- Nguyen, H. X. e Thiran, P. (2004). Active measurement for multiple link failures diagnosis in ip networks. In *Passive and Active Network Measurement*, p. 185–194. Springer.
- Paul Tune, M. R. (2013). Internet Traffic Matrices. In Haddadi, H. e Bonaventure, O., editors, *Recent Advances in Networking*, chapter 3. ACM SIGCOMM eBook.
- Sandvine (2014). Global Internet Phenomena Report. <https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/1h-2014-global-internet-phenomena-report.pdf>.
- Suh, K., Guo, Y., Kurose, J., e Towsley, D. (2006). Locating network monitors: complexity, heuristics, and coverage. *Computer Communications*, 29:1564–1577.
- Wu, Q., Li, Z., Yang, J., Xie, G., e Salamatian, K. (2012). Efficient traffic flow measurement for isp networks. In *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, p. 348–351. IEEE.
- Zhang, Y., Roughan, M., Willinger, W., e Qiu, L. (2009). Spatio-temporal compressive sensing and internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 39:267–278.
- Zhao, Q. G., Kumar, A., Wang, J., e Xu, J. J. (2005). Data Streaming Algorithms for Accurate and Efficient Measurement of Traffic and Flow Matrices. *SIGMETRICS Perform. Eval. Rev.*, 33(1): 350–361.