# Reordenando Assinaturas em Mecanismos de Inspeção de Pacotes Baseado em Prioridade Dinâmica

**Petrônio Júnior, Wesley Melo, Rafael Antonello, Stenio Fernandes, Djamel Sadok**

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

{pglj,wdbm,rta,sflf,jamel}@cin.ufpe.br

***Abstract.*** *Traffic Classification and Identification plays a key role in many network management activities. In this context, DPI is one of the most used methods, being very accurate. However, a DPI system has high computational cost. Many components of a DPI architecture are constantly studied from the point of view of their impact on the computational performance of the system, in special, the signature set. In this work, the impact on the order of signatures is attested through a static ordering on the signature list. Additionally, a method to order dynamically the signature set is proposed, reducing the processing time of the DPI. The results demonstrate the impact of the order of signatures on the performance of the DPI, as well as the performance gains (through dynamics ordering) of more than 65% of the processing time. Finally, the proposed method in this work can be combined with other state-of-the-art techniques to achieve a better DPI performance.*

***Resumo.*** *Classificação e identificação de tráfego tem um papel fundamental em diversas atividades do gerenciamento de redes de computadores. Nesse contexto, DPI é um dos métodos mais utilizados, além de ser bastante preciso. No entanto, um sistema DPI apresenta um alto custo computacional. Vários componentes da arquitetura de um DPI são constantemente estudados do ponto de vista do seu impacto no desempenho computacional do sistema, em especial a base de assinaturas. Neste trabalho, o impacto da ordem das assinaturas é atestado através de uma ordenação estática da lista de assinaturas. Adicionalmente, foi proposto um método de ordenação dinâmica com o intuito de reduzir o tempo de processamento da inspeção de pacotes. Os resultados demonstraram a influência da ordem das assinaturas no desempenho do DPI, bem como o ganho de desempenho, através da ordenação dinâmica, de mais de 65% do tempo de processamento. Por fim, o método proposto neste trabalho pode ser combinado com outras técnicas do estado da arte para alcançar um melhor desempenho do DPI.*

## 1. Introdução

Diversas áreas relacionadas a redes de computadores ratificam a importância da identificação de tráfego passante na rede. Como exemplo, é possível citar áreas como gerenciamento de redes [Song e Zhou 2013] e detecção de anomalias [Chandola et al. 2009]. De maneira geral, existem três métodos para classificação de tráfego apresentados na literatura [Callado et al. 2009]: classificação baseada em portas, classificação baseada em fluxos e classificação baseada em pacotes. Existem diversos trabalhos utilizando as técnicas apresentadas na literatura. No entanto, é possível

destacar a Inspeção Profunda de Pacotes (DPI) – que é baseada em pacotes – como a técnica mais precisa do ponto de vista da classificação. Em outras palavras, esse método alcança as maiores taxas de pacotes corretamente classificados, inspecionando o conteúdo dos pacotes que passam na rede monitorada [Callado et al. 2009]. Nesse caso, o método inspeciona a carga útil do pacote à procura de um determinado padrão, denominado assinatura. Para que a classificação seja efetiva, é necessário que várias assinaturas sejam procuradas na carga útil do pacote até que algum padrão seja encontrado (casamento). Embora seja um método preciso, o DPI enfrenta alguns problemas de desempenho relacionados ao seu alto custo computacional [Santos et al. 2013]. Alguns estudos propõem formas de melhorar o desempenho do DPI, como a utilização de GPUs para incrementar sua capacidade de processamento [Santos et al. 2013] ou a inspeção apenas dos pacotes iniciais de um fluxo [Fernandes et al.]. Melhorar o desempenho computacional de um sistema DPI permanece sendo uma questão em aberto na comunidade científica.

Diante da revisão de literatura realizada, é possível perceber que nenhuma pesquisa anterior analisa a ordem das assinaturas como um importante elemento que possa interferir no desempenho de um DPI. [Antonello et al. 2011] avaliaram o impacto de diferentes listas de assinaturas em um DPI, mas não foi analisada a ordem das assinaturas em uma mesma lista. Ordenar as assinaturas dinamicamente para melhorar o desempenho de um DPI não é uma tarefa trivial. Pesquisas em outras áreas de conhecimento propuseram abordagens similares para os problemas estudados, como algoritmos de cache [Gao et al. 2013] e algoritmos de *placement* de servidores [Huang et al. 2012]. Embora o cenário de aplicação seja distinto, a ideia é bastante similar. No caso da ordenação dinâmica das assinaturas, a ideia consiste basicamente na priorização das assinaturas mais relevantes para a classificação. Isso é, considerando que as assinaturas estão em uma lista que é percorrida sequencialmente, as assinaturas mais relevantes são posicionadas no início da lista. Essa técnica permite que os casamentos ocorram mais rápido e o sistema DPI economize tempo de processamento. Caso toda a lista precise ser percorrida, um maior tempo de processamento será gasto. Para um único caso, a diferença entre os dois cenários citados será irrelevante. No entanto, em um cenário de redes *multigigabits*, esse evento ocorre milhões de vezes em um dia, tornando significante o tempo de processamento poupado.

Neste trabalho, foi realizado um estudo minucioso do impacto que a ordenação das assinaturas utilizadas por um sistema DPI causa ao seu desempenho. Nesse caso, evidências quantitativas desse impacto são apresentadas. Adicionalmente, um novo mecanismo foi proposto e desenvolvido para alterar dinamicamente a ordem das assinaturas na lista a fim de melhorar o desempenho do sistema. Em resumo, este trabalho apresenta duas contribuições: (i) demonstração do impacto da ordem das assinaturas no desempenho do DPI e (ii) desenvolvimento de um novo método de ordenação dinâmica das assinaturas.

O restante deste trabalho é organizado da seguinte forma: Na Seção 2, os trabalhos relacionados são apresentados. Um novo método de ordenação é proposto na Seção 3. Em seguida, na Seção 4 a metodologia é descrita e os resultados são apresentados na Seção 5. Por fim, as conclusões e trabalhos futuros são expostos.

## 2. Trabalhos Relacionados

Ao longo da última década, vários estudos foram realizados a fim de avaliar o desempenho computacional de um sistema DPI. Entretanto, as pesquisas abordam diferentes aspectos do sistema para alcançar seus resultados. É importante destacar que a maioria dessas técnicas são complementares e podem ser aplicadas em conjunto a fim de alcançar melhores resultados.

Diversas pesquisas nessa área estão relacionadas à compressão de Autômatos Finitos Determinísticos (DFA). Em [Yu et al. 2006], foi proposto o agrupamento de diferentes autômatos. Em outras palavras, um único autômato é utilizado para representar um conjunto de autômatos (ou um subconjunto dos autômatos contidos em um sistema DPI). Em comparação com outros trabalhos da literatura, os resultados obtidos pelos autores demonstraram ganhos de até 700 vezes a velocidade de processamento original. [Antonello et al. 2012] demonstrou que o RCDFA (Ranged Compressed Deterministic Finite Automaton) comprime transições sem adição relevante de acessos à memória. Cerca de 97% de redução em espaço em memória utilizado foi alcançado em comparação ao DFA original e aproximadamente 93% em relação a técnicas existentes. Outra pesquisa nessa área foi proposta por [Becchi e Crowley 2013], em que surge a ideia do A-DFA (Amortized time-bandwidth overhead DFA). Quando comparado ao proposto em D²FA [Kumar et al. 2006], o A-DFA alcança a mesma taxa de compressão com menos consumo de memória. Além disso, os autores realizaram um estudo de viabilidade da combinação do algoritmo proposto com redução de alfabeto e DFA *multistride*. Por fim, também foram discutidas por [Becchi e Crowley 2013] diferentes codificações de memória. Em todas as pesquisas apresentadas até este ponto, é possível observar que o foco principal está na economia de memória.

Outras partes do sistema DPI foram modificadas por diversos trabalhos a fim de alcançar melhorias de performance. Considerando a velocidade de casamento, o trabalho apresentado por [Santos et al. 2013] apresentou a arquitetura FBTI (Flow-based Traffic Identification). Essa abordagem propõe a utilização de uma GPU para aumentar a capacidade de processamento e, assim, acelerar a velocidade de casamento de um sistema DPI. Os autores desse trabalho utilizaram apenas os bytes iniciais de cada fluxo na classificação para otimização ainda maior do desempenho do sistema. No cenário de melhor caso, a arquitetura proposta alcançou uma vazão de aproximadamente 120 Gbps.

Outra técnica bastante interessante é proposta em [Fernandes et al. 2009]. Nela, os autores propuseram que a inspeção de apenas os 7 primeiros pacotes de cada fluxo seria suficiente para classificar de maneira satisfatória o tráfego e, em contrapartida, reduziria o tempo de processamento. Os resultados demonstraram um decréscimo de aproximadamente 75% no tempo de processamento em comparação à abordagem em que todos os pacotes do fluxo são inspecionados.

Por fim, o arcabouço proposto por [Antonello et al. 2011] analisa o impacto do uso de diferentes bases de assinaturas, avaliando e comparando suas complexidades. Os resultados expostos se baseiam na execução de um mesmo DPI com conjuntos de assinaturas diferentes. Nesse contexto, os autores concluíram que não é justa a comparação do desempenho de um DPI utilizando bases de assinaturas de diferentes complexidades.

Em resumo, é possível perceber que os trabalhos da literatura nessa área, de maneira geral, estão relacionados à compressão de autômatos, à plataforma utilizada pelo sistema DPI e à complexidade de diferentes listas de assinaturas. Nenhum estudo anterior analisou a ordem das assinaturas na base. Em outras palavras, de acordo com o levantamento realizado na literatura, a abordagem proposta neste trabalho é a primeira a analisar o conteúdo da base de assinaturas de um DPI e obter um melhor desempenho com sua ordenação.

## 3. Prioridade no conjunto de assinaturas

Diante do objetivo de sistemas DPI de encontrar assinaturas de aplicações na carga útil de pacotes, existem várias formas de representar esses padrões. Uma maneira bastante utilizada é o formato de expressões regulares (regex). Neste formato, podem ser observadas algumas listas de assinaturas que são disponibilizadas pela comunidade científica. Uma das mais populares é a lista de assinaturas do L7-Filter[1].

A importância da lista de assinaturas fica evidente quando se estuda minuciosamente o processo de classificação de um DPI. No cenário proposto neste trabalho, os pacotes chegam ao sistema em elevadas taxas, podendo chegar à ordem de *Gigabits* por segundo. O DPI armazena o seu conjunto de assinaturas previamente processadas (expressões regulares representadas como DFAs) na memória principal utilizando uma lista ligada. O processo de classificação de um pacote pode ser sintetizado da seguinte forma: uma vez que o pacote chega à máquina de medição onde a análise é executada, o DPI examina a carga útil do pacote buscando sequencialmente um padrão correspondente na lista de assinaturas. Caso esse padrão seja identificado, ocorre o casamento, que significa que o pacote (e consequentemente o fluxo associado) foi classificado. Quando um casamento ocorre, o sistema não realiza mais comparações para o fluxo associado ao pacote, encerrando seu processamento para o fluxo em questão. Nesse cenário, quanto antes ocorrer o casamento do pacote com a assinatura, mais cedo a análise cessará e o tempo de processamento será reduzido. Dessa forma, o cenário ideal se configura quando a assinatura do pacote analisado está posicionada nas primeiras posições da lista.

Inicialmente, uma solução interessante seria realizar a ordenação prévia da lista de assinaturas antes mesmo que o sistema DPI inicie sua execução. No entanto, para que isso fosse possível, seria necessário conhecer previamente quais aplicações (ou mesmo classes de aplicações) são responsáveis pela maior parcela do tráfego da rede. O benefício dessa abordagem fica evidenciado quando as características da lista de assinaturas são observadas com mais detalhes. São 123 assinaturas presentes na lista do L7-Filter [Antonello et al. 2011], que é a lista utilizada neste trabalho. Alguns agrupamentos interessantes podem ser observados a cerca das assinaturas na Tabela 1.

É interessante notar que cerca de 15% do total de assinaturas pertence ao grupo "Jogos". Dessa forma, se o sistema conhece a frequência com que as aplicações aparecem no tráfego, é possível posicionar o grupo "Jogos" no início da lista, no final ou mesmo retirá-lo do conjunto de assinaturas de acordo com o interesse do administrador do sistema no resultado do DPI. Entretanto, esse tipo de informação dificilmente está disponível de maneira prévia para o sistema DPI. Além disso, a

[1] http://l7-filter.sourceforge.net/. Acessado em Dezembro de 2013.

inspeção de pacotes costuma ocorrer em tempo real e uma eventual mudança do perfil do tráfego inspecionado não seria percebida. De acordo com [Cho et al. 2006], o perfil de tráfego da Internet muda à medida que o tempo passa. Essas mudanças ocorrem em diferentes escalas e dificilmente são previsíveis.

Sob a ótica da lista de assinaturas, a capacidade de adaptação às mudanças de tráfego é indispensável para sistemas DPI de alto desempenho. Por exemplo, a adaptação da lista de assinaturas pode melhorar o desempenho computacional de um DPI em um cenário de *flash crowd*. De maneira geral, um *flash crowd* pode ser definido como um fenômeno social traduzido na Internet que aumenta o volume de uma parcela específica do tráfego da rede rapidamente e é bastante difícil prevê-lo [Hiremagalore et al. 2013] [Zhanikeev et al. 2013]. Se o novo comportamento da rede for percebido pelo DPI e as assinaturas relevantes para o evento forem posicionadas no topo da lista de assinaturas, um grande volume de processamento será economizado.

**Tabela 1. Assinaturas e possíveis grupos para o conjunto de assinaturas do L7-Filter**

| Nome do grupo | Assinaturas |
|---|---|
| HTTP | http; http-rdsp; http-dap; http-freshdownload; http-itunes; httpaudio; httpcachehit; httpcachemiss; httpvideo; quicktime |
| P2P | 100bao; applejuice; ares; bittorrent; directconnect; edonkey; fasttrack; gnucleuslan; gnutella; goboogy; hotline; imesh; kugoo; mute; napster; openft; poco; pplive; soribada; soulseek; tesla; thecircle; xunlei |
| SNMP | snmp; snmp-mon; snmp-trap |
| Mensagem instantânea | aim; aimwebcontent; gtalk; irc; jabber; msnmessenger; qq; yahoo; |
| Jogos | armagetron; battlefield2; battlefield1942; battlefield2142; counterstrike-source; dayofdefeat-source; doom3; guildwars; halflife2-deathmatch; liveforspeed; mohaa; quake-halflife; quake1; runesofmagic; subspace; teamfortress2; worldofwarcraft; xboxlive |

A solução proposta neste trabalho está relacionada às estatísticas de casamento de assinaturas e aos custos definidos para cada assinatura. Essa informação associada de maneira específica fornece subsídios para que existam critérios de priorização na lista de assinaturas. Esse critério é definido de acordo com a Equação 1:

**Equação 1. Função de prioridade da assinatura *sig***

$$\mathcal{F}_{sig} = \frac{P_{sig}}{C_{sig}}$$

Nessa equação, a função de prioridade ($F_{sig}$) de uma assinatura *sig* é definida pela razão entre sua popularidade ($P_{sig}$) e o seu custo ($C_{sig}$). A popularidade de uma assinatura é a parcela mutável de sua função de prioridade. Cada casamento proporcionado por uma assinatura específica incrementa uma unidade à sua popularidade. Já o custo é um valor fixo e particular a cada assinatura. A definição do custo de cada assinatura será detalhada na subseção a seguir. Em linhas gerais, a ideia

da priorização das assinaturas está exposta na Figura 1. Essa figura mostra em dois momentos distintos um possível cenário da lista de assinaturas e como a função de prioridade realiza a reordenação da lista.

No primeiro momento ($t_1$), cada assinatura tem um valor de função associado. O próximo pacote que chega ao sistema DPI pertence a uma aplicação (e.g. DOOM) e promove um casamento com a respectiva assinatura. Nesse cenário, o valor da função f2 é atualizado. Em seguida, o valor de f2 é comparado ao valor da função da assinatura acima na lista (f1). Dado que o resultado da função de prioridade da aplicação DOOM é maior que o da função do BitTorrent, ocorre a inversão na ordem das assinaturas envolvidas na comparação. A partir dessa troca, o cenário observado passa a ser o de $t_2$. Essa operação ocorre porque, de acordo com a informação fornecida pela função de prioridade, espera-se que a o tempo de processamento seja menor com a nova ordem das assinaturas.

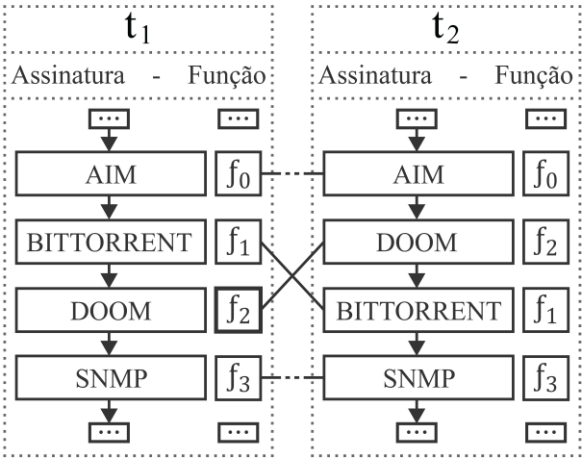


**Figura 1. Funcionamento da função de prioridade**

Alguns outros elementos poderiam ser inseridos nessa função e, com maior eficácia, as assinaturas poderiam ser ordenadas. Um exemplo disso é a utilização de uma janela de tempo para controle da popularidade, corrigindo, assim, possíveis desvios na ordenação. Entretanto, é necessário que o mecanismo de ordenação cause a menor sobrecarga possível ao processamento do DPI. No nosso cenário, o sistema DPI opera em redes de alta velocidade e, portanto, deve ser suficientemente rápido para tratar os pacotes que chegam à interface de rede. Além disso, o benefício da reordenação das assinaturas deve superar a sobrecarga gerada pela função de prioridade. Algoritmos avançados de ordenação apresentam alta complexidade [Cormem et al. 2001] e não se adéquam à realidade de um DPI. Em redes de alta velocidade, o DPI deve operar na escala de nanossegundos.

### 3.1. Custo da assinatura

Como citado anteriormente, cada assinatura tem um custo associado. O cálculo desse custo é previamente realizado (antes do início da operação do DPI) e o custo obtido se mantém fixo durante todo o funcionamento do sistema e para suas próximas execuções.

Uma vez que cada assinatura é uma expressão regular e é representada por um DFA, o custo foi estabelecido como sendo o menor caminho desde o estado inicial do autômato até um estado de aceitação. Em outras palavras, o custo é definido como o

número de saltos desde o estado inicial até o estado de aceitação do DFA que representa a assinatura. Para a obtenção do custo, foram gerados os autômatos que representam cada assinatura e um algoritmo de busca em profundidade foi implementado para identificar esse valor.

Já que o custo é um valor definido antes da execução do DPI, esse cálculo pode ser tão complexo quanto necessário, sem adicionar prejuízo ao tempo de processamento do DPI. Além disso, outras métricas podem ser utilizadas para definir o custo. Por exemplo, o maior caminho até o estado de aceitação, o número de estados do autômato ou o seu número de transições.

## 4. Metodologia

Neste trabalho, o impacto da ordenação das assinaturas sobre o processo de classificação de um DPI foi analisado. Nesse caso, o objetivo é reduzir o tempo necessário para classificar todos os fluxos de um *trace.* Portanto, a análise realizada neste trabalho avalia o tempo de processamento e não a precisão do DPI. Dessa forma, o número de fluxos que são analisados é mais relevante que o tipo de tráfego dos *traces* utilizados.

Para a avaliação da abordagem proposta, poderiam ser utilizados *traces* reais ou sintéticos. No entanto, um tráfego real possui características específicas, apresentando variações que um tráfego sintético não consegue reproduzir. Nesse caso, foram utilizados três *traces* reais.

Dois dos *traces* (Trace1 e Trace2) foram coletados em um *backbone* do Brasil. Esses *traces* apresentam um grande volume de pacotes e fluxos. A terceira coleta de tráfego (Trace3) foi realizada através da ferramenta GT e está disponível publicamente [Gringoli et al. 2009]. O *trace* coletado pelo GT consiste na concatenação da coleta de dois dias consecutivos.

Já que o tipo de tráfego especificamente não interfere diretamente no desempenho da abordagem proposta, os experimentos realizados podem ser reproduzidos considerando as informações contidas na Tabela 2.

**Tabela 2. *Traces* e características**

| *Trace* | # de pacotes | # de fluxos | Pacotes inspecionados | Número de aplicações |
|---|---|---|---|---|
| Trace1 | 118308001 | 838975 | 3429920 | 31 |
| Trace2 | 114168550 | 871055 | 3519427 | 33 |
| Trace3 | 7512639 | 74447 | 778829 | 16 |

Para avaliar o impacto da ordem das assinaturas, duas abordagens foram utilizadas. Na primeira, são realizadas mudanças estáticas na ordem das assinaturas do L7-Filter antes da execução do DPI. Nesse caso, foram definidos 5 arranjos diferentes para a avaliação das mudanças estáticas: Ordenada (assinaturas em ordem alfabética), Inversa (ordem inversa à lista Ordenada), Aleatória 1, Aleatória 2 e Aleatória 3. O conjunto de assinaturas Ordenada é considerado como o caso base da lista de assinaturas. A segunda abordagem trata da ordenação dinâmica, realizada através da função de prioridade. É interessante notar que a ordenação dinâmica foi realizada

partindo das 5 diferentes organizações realizadas estaticamente. Essa avaliação é importante porque um ponto de partida (ordenação inicial) diferente poderia impactar no tempo de convergência da ordenação dinâmica.

Para analisar a efetividade da ordenação dinâmica, a avaliação utiliza como base de comparação o desempenho do DPI sobre a lista estática de assinaturas. Em outras palavras, cada uma das listas de assinaturas (Ordenada, Inversa, Aleatória 1, Aleatória 2, Aleatória 3) é analisada com e sem a utilização da função de prioridade.

A ideia deste trabalho é avaliar e apresentar ganhos de desempenho computacional de um sistema DPI. Dessa forma, os valores absolutos do desempenho do DPI serão apresentados apenas para a lista Ordenada, que é a base de comparação. Nesse cenário, a métrica utilizada é o tempo total necessário para analisar todo o "caminho" percorrido pelo pacote dentro do sistema, denominada tempo de processamento do pacote. Essa métrica considera também o tempo gasto para a ordenação, quando houver, das assinaturas. A fim de ter uma visão geral do sistema, o tempo de processamento de cada pacote é agregado em um nível menor de granularidade, no nível de fluxos. Essa métrica será tratada apenas como tempo de processamento e será o tempo total de processamento dos fluxos por cenário. A avaliação foi realizada de maneira *offline* para que sejam evitadas perdas de pacotes e, consequentemente, medições injustas entre diferentes cenários.

Todos os testes foram executados em uma máquina Linux (Ubuntu 12.04 - x86), kernel 3.11, Intel (R) Core (TM) i7-2600 3.4 GHz, 8 GB RAM.

## 5. Resultados e Discussão

Inicialmente, os resultados foram obtidos através dos experimentos utilizando a lista de assinaturas Ordenada, tanto para a ordenação estática quanto para a ordenação dinâmica. Como exposto na Seção 4, todas as comparações subsequentes serão valores relativos à base de comparação. Na Tabela 3, os valores absolutos para a lista Ordenada são apresentados para que a comparação com outros trabalhos seja possível.

**Tabela 3. Tempo de processamento para a lista de assinaturas Ordenada**

| Sem função de prioridade | | Com função de prioridade | |
|---|---|---|---|
| **Lista Ordenada** | **Tempo de processamento (ns)** | **Lista Ordenada** | **Tempo de processamento (ns)** |
| Trace1 | 7,509E+11 | Trace1 | 4,001E+11 |
| Trace2 | 7,857E+11 | Trace2 | 5,292E+11 |
| Trace3 | 1,812E+11 | Trace3 | 3,053E+11 |

Nesse caso, é possível perceber que os dois primeiros *traces* apresentam um tempo de processamento menor quando a função de prioridade é utilizada. No entanto, para o Trace3, ocorre um aumento nesse tempo de processamento, que pode ser justificado pelo menor número de fluxos presente nesse tráfego coletado. Nesse caso, a sobrecarga gerada pela função de prioridade não é superada pelos benefícios da ordenação.

Na Tabela 4, são apresentados os percentuais de diferença do processamento com e sem função de prioridade, que expõem a relação entre os valores absolutos que foram apresentados na tabela anterior.

**Tabela 4. Ganho percentual do tempo de processamento com o uso da função de prioridade**

| *Trace* utilizado | Tempo de processamento economizado (%) |
|---|---|
| Trace1 | 43 |
| Trace2 | 33 |
| Trace3 | -68 |

É possível perceber que, para o Trace1 e para o Trace2, o tempo de processamento apresenta ganho de desempenho quando a função de prioridade é utilizada, sendo cerca de 43% no melhor caso (Trace1). Entretanto, existe uma queda de desempenho quando a função é aplicada ao cenário do Trace3. Como dito anteriormente, isso ocorre por conta do baixo número de fluxos presentes nesse *trace.* Nesse caso, menos interações ocorrem, sendo insuficiente para que a lista de assinaturas seja ordenada de maneira ótima. Não há fluxos suficientes para que a função de prioridade exerça seus benefícios, superando a sobrecarga de sua operação. No entanto, em um cenário real de redes de altas velocidades, a quantidade de fluxos será consideravelmente maior que o observado no Trace3. Dessa forma, os resultados a partir deste ponto serão limitados às análises envolvendo o Trace1 e o Trace2, já que, por conta do seu número de fluxos, permitem que a função de prioridade seja efetiva.

## 5.1. Ordenação estática

A Figura 2 apresenta o tempo proporcional que cada ordenação diferente leva para que todo o *trace* seja inspecionado pelo DPI.

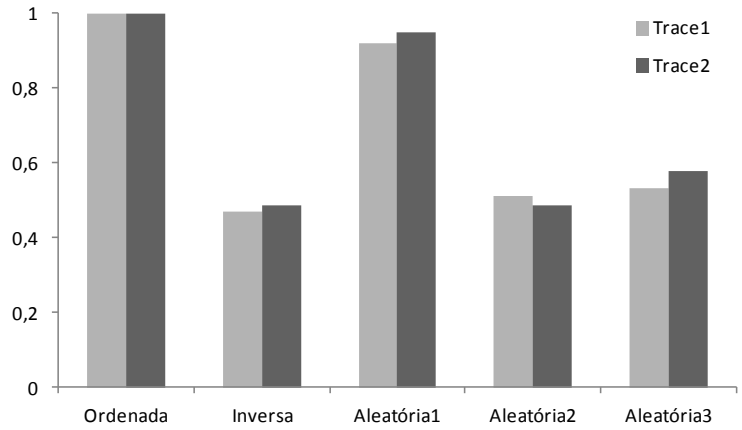


**Figura 2. Valor relativo do tempo de processamento para ordenações estáticas**

Para ambos os *traces* apresentados nessa figura, a base de comparação é o desempenho obtido com a lista de assinaturas Ordenada. Coincidentemente, para os dois casos a lista Ordenada apresenta o maior tempo de processamento. A lista Inversa apresenta os melhores resultados, sendo aproximadamente 0,47 do desempenho da base Ordenada para o Trace1 e aproximadamente 0,48 para o Trace2. Para o Trace1, a lista Aleatória1 apresenta 0,92 em relação à lista Ordenada e, para o Trace2, 0,92 também em relação a sua respectiva base de comparação.

A análise do desempenho de diferentes ordens de assinaturas realizadas estaticamente visa a comprovação da hipótese de que essa ordem afeta o desempenho

do sistema. Como é possível observar, a ordem inicial das assinaturas impacta, de fato, no desempenho do DPI. O desempenho das listas aleatórias demonstrou que, em alguns casos, é possível ser tão eficaz quanto o melhor caso (Inversa) e tão custoso quanto o pior caso (Ordenada). Dessa forma, fica evidente que a ordem das assinaturas impacta diretamente no desempenho do sistema DPI. Portanto, ordenar a lista de assinaturas é uma abordagem interessante para melhorar o desempenho da inspeção de pacotes. Embora esteja fora do escopo deste trabalho encontrar outras ordenações aleatórias eficientes, é provável que existam outras ordenações que permitam um menor tempo de processamento do DPI. Além disso, alguns mecanismos de ordenação podem ser propostos considerando a complexidade de cada assinatura ou suas inter-relações.

Adicionalmente, algumas otimizações podem ser associadas à ordenação estática das assinaturas. Quando apresentam características similares, as assinaturas podem ser agrupadas, reduzindo o número de assinaturas que precisam ser analisadas. Além disso, caso o perfil de tráfego seja conhecido, uma assinatura indesejada pode ser removida da lista, permitindo uma menor sobrecarga ao tempo de processamento.

## 5.2. Ordenação dinâmica

Para ratificar a relevância da ordenação dinâmica, os resultados expostos a seguir comparam os tempos de processamento com e sem o uso da função de prioridade. Para todos os casos, como dito anteriormente, o tempo de processamento base é o da lista Ordenada sem utilizar a função de prioridade. Todos os outros valores são relativos a essa base. Na Figura 3, são apresentados os tempos de processamento relativos obtidos quando o Trace1 foi inspecionado.

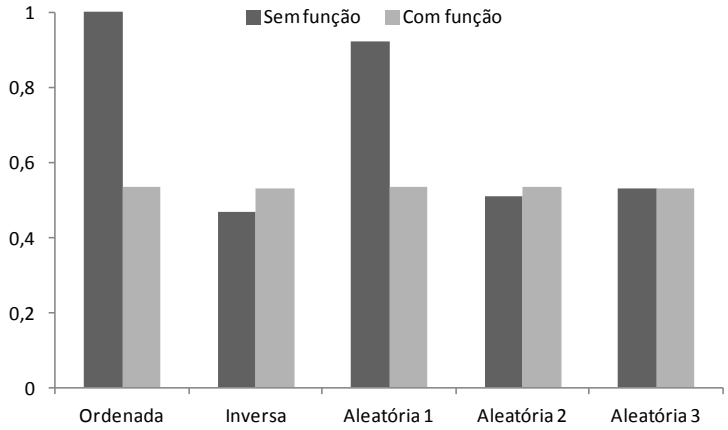


**Figura 3. Tempos de processamento relativos para o Trace1**

É possível observar que os resultados obtidos com o uso da função de prioridade superam amplamente os obtidos com as listas Ordenada e Aleatória 1 sem a função. Nesses casos, contata-se que, apesar da sobrecarga gerada pela função de prioridade, a ordenação das assinaturas na lista permite que o tempo de processamento seja reduzido. Entretanto, quando o comportamento das outras listas é observado, percebe-se que não há melhora no tempo total de processamento. Diante desse cenário, o tempo de processamento utilizando a função de prioridade foi analisado em duas partes: (i) o tempo de classificação propriamente dito e (ii) a sobrecarga gerada pela operação da função. A sobrecarga gerada pela função de prioridade e pelas trocas de posição das assinaturas representa em média aproximadamente 42,5% do tempo de processamento

para o Trace1. Nesse caso, a reordenação proporcionada pela função de prioridade demonstra um desempenho bastante superior ao alcançado sem a função, como pode ser observado na Figura 4.

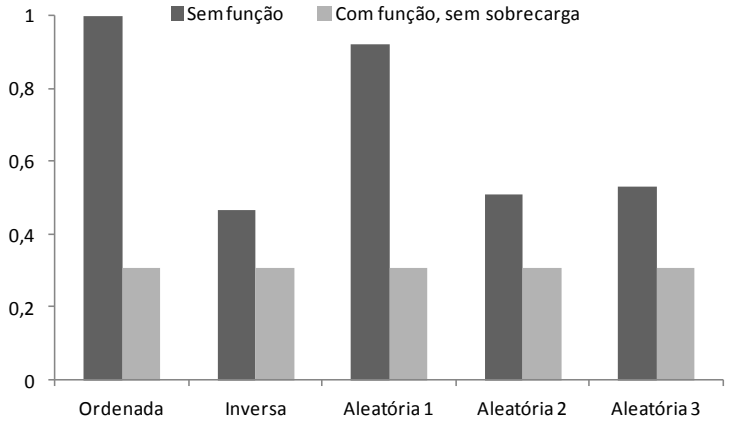


**Figura 4. Tempos de processamento relativos para o Trace1 desconsiderando a sobrecarga**

Em seguida, observando a Figura 5 e a Figura 6, percebe-se que o cenário propiciado pelo Trace2 é bastante similar ao descrito acima para o Trace1. Na Figura 5, existe um ganho considerável de desempenho quando são utilizadas as listas Ordenada e Aleatória 1. Nesse cenário, o tempo de processamento para as listas Inversa, Aleatória 2 e Aleatória 3 apresenta uma perda significativa de desempenho quando a função é utilizada.

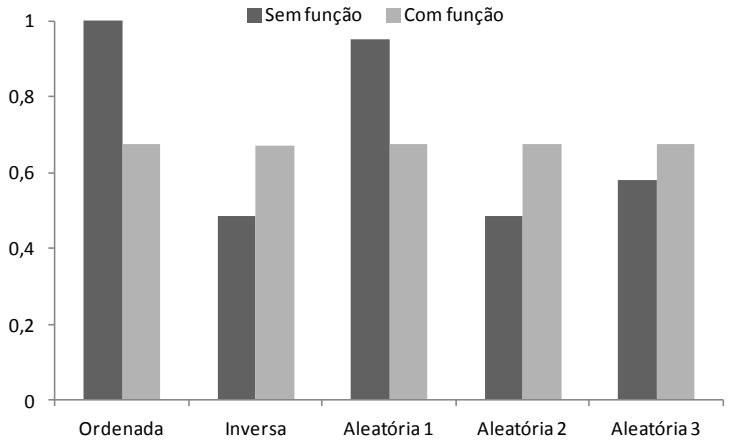


**Figura 5. Tempos de processamento relativos para o Trace2**

Mais uma vez, a sobrecarga decorrente do cálculo da função e das trocas de posição das assinaturas representa uma grande parcela do tempo de processamento total obtido. Para o Trace2, o valor médio da sobrecarga obtido é cerca de 49% do tempo de processamento do método que usa a função de prioridade. Quando a sobrecarga é excluída do tempo de processamento, os benefícios da ordenação proporcionada pelo método utilizado são bastante relevantes. Na Figura 6, são apresentados os tempos de processamento sem sobrecarga. Assim como no cenário anterior, a ordenação das assinaturas de acordo com a função de prioridade sem a sobrecarga tem um tempo de processamento consideravelmente menor que o obtido sem a função.

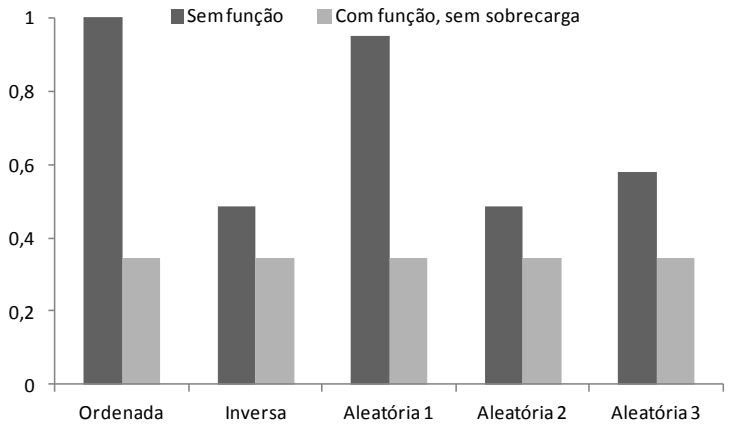


**Figura 6. Tempos de processamento relativos para o Trace1 desconsiderando a sobrecarga**

É interessante destacar que um trabalho voltado à otimização da função ou da estrutura de dados pode permitir que a sobrecarga seja minimizada e, com isso, o tempo de processamento total seja reduzido. Sem a sobrecarga, o desempenho da ordenação proporcionada pela função é significativamente superior ao alcançado sem a função de prioridade. Essa sobrecarga pode ser reduzida, por exemplo, com a utilização de uma estrutura de dados otimizada, própria para executar trocas de posições entre as assinaturas da lista.

Por fim, o uso da função de prioridade apresenta uma característica bastante interessante e desejável para o cenário estudado neste trabalho. Independente da ordem inicial da lista de assinaturas, o tempo de processamento utilizando a função converge para o mesmo desempenho. Isso pode ser observado na Figura 7, em que a variação do tempo de processamento para cada *trace* é exposta.

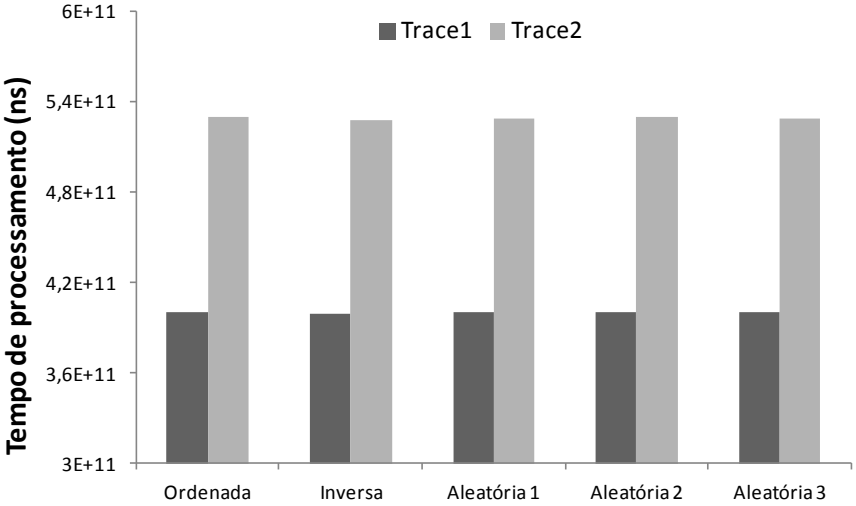


**Figura 7. Os tempos de processamento utilizando a função de prioridade se mantêm constantes**

Essa característica é bastante interessante já que, mesmo que não apresente o melhor desempenho para qualquer classificação, o tempo de processamento será baixo e estável. Nesse caso, percebe-se que o desempenho se mantém constante para as diferentes listas de assinaturas utilizadas.

## 6. Conclusão e Trabalhos Futuros

Neste trabalho, foi realizada uma análise de como a ordem das assinaturas pode afetar um sistema DPI. A aleatoriedade foi aplicada à lista inicial de assinaturas e, com isso, foi possível perceber que foram alcançados tempos de processamento próximos ao melhor caso e também ao pior caso. Isso demonstra que a maneira em que estão organizadas as assinaturas impacta diretamente do desempenho do sistema.

Além disso, foi proposto um método de ordenação dinâmica das assinaturas de acordo com os pacotes que chegam e são inspecionados. A ideia da proposta é criar um método de ordenação capaz de adaptar a lista de assinaturas de maneira ótima de acordo com o tráfego que está sendo analisado. Para que fosse possível realizar a ordenação dinâmica, um critério de priorização foi definido através da função de prioridade. Essa função leva em consideração a popularidade de uma assinatura e seu custo. Uma vez definida a técnica de ordenação, uma análise do tempo de processamento do método proposto foi avaliada. Os resultados demonstraram a efetividade da utilização da função, principalmente quando a sobrecarga do processo é retirada.

Adicionalmente foram discutidas algumas possíveis otimizações que podem ser aplicadas a listas de assinaturas estaticamente. Em tempo, essas otimizações podem ser combinadas com a ordenação dinâmica. Uma possível melhoria está relacionada ao agrupamento de assinaturas similares ou mesmo a remoção de assinaturas da lista (quando não interessam à análise do DPI).

Outras otimizações podem ser aplicadas à ordenação dinâmica, como a alteração do cálculo do custo ou a temporização da função de prioridade (os valores seriam mantidos apenas durante uma janela de tempo). No entanto, as otimizações envolvendo a função de prioridade devem observar a complexidade das operações envolvidas já que a sobrecarga gerada pela função deve ser superada pelos seus benefícios. Além de otimizações, outras funções podem ser propostas, estabelecendo um novo critério de prioridade. Mais uma vez, deve ser observada a complexidade da função que será implementada. É essencial destacar que a função de prioridade proposta neste trabalho pode ser combinada com outras técnicas existentes na literatura. Dessa forma, não há impedimentos para que uma solução combinada seja utilizada visando a melhoria do desempenho de um sistema DPI.

Este trabalho foi o primeiro a comprovar e analisar os impactos da ordem das assinaturas no tempo de processamento de um DPI. Diante do levantamento de literatura realizado, nenhuma outra pesquisa anterior considerou a lista de assinaturas como um fator relevante para o desempenho do sistema. Por fim, já que esta pesquisa permite diversos desmembramentos, alguns trabalhos futuros podem ser destacados como a criação de novas funções de prioridade, o estudo da complexidade dessas funções e a análise das correlações entre assinaturas.

## Referências

Antonello, R. et al., (2011) "Characterizing signature sets for testing DPI systems," GLOBECOM Workshops (GC Wkshps), IEEE.

Antonello, R. et al., (2012) "Deterministic Finite Automaton for scalable traffic identification: The power of compressing by range," Network Operations and Management Symposium (NOMS), IEEE April 2012.

Becchi, M. and Crowley, P., (2013) “A-DFA: A Time- and Space-Efficient DFA Compression Algorithm for Fast Regular Expression Evaluation”, ACM Trans. Archit. Code Optim. 10, 1, (April 2013).

Callado, A. et al., (2009) "A Survey on Internet Traffic Identification," Communications Surveys & Tutorials, IEEE.

Chandola, V. et al. (2009) “Anomaly detection: A survey”. ACM Comput. Surv. 41, 3, Article 15 (July 2009).

Cho, K. et al. (2006) “The impact and implications of the growth in residential user-to-user traffic”. SIGCOMM Comput. Commun. Rev. Aug 2006.

Cormem, T. H. et al. (2001). Introduction to algorithms. MIT press, Cambridge.

Fernandes, S. et al. (2009) "Slimming Down Deep Packet Inspection Systems," INFOCOM Workshops 2009, IEEE , vol., no., pp.1,6, 19-25 April 2009.

Gao, Y. et al., (2013) "A Cache Management Strategy for Transparent Computing Storage System." Trustworthy Computing and Services. Springer Berlin Heidelberg, 2013. 651-658.

Gringoli, F. et al., (2009) “GT: picking up the truth from the ground for Internet traffic”, ACM SIGCOMM Comp. Comm. Review, Oct. 2009.

Hiremagalore, S. et al. (2013) "Improving network response times using social information", Social Network Analysis and Mining (2013): 1-12.

Huang, X. et al., (2012) "A Novel Replica Placement Strategy for Data Center Network", ICITMS 2012, Proc. Of., Springer Berlin.

Kawano, S. et al., (2012) "High-Speed DPI Method Using Multi-Stage Packet Flow Analyses," 9th APSITT 2012 Nov. 2012.

Kumar, S. et al., (2006) “Algorithms to accelerate multiple regular expressions matching for deep packet inspection”, ACM SIGCOMM Conf. on ATAPCC.

Liu, D. et al., (2013) "Network traffic anomaly detection using clustering techniques and performance comparison", 26th IEEE CCECE, 2013.

Liu, T. et al., (2011) "An efficient regular expressions compression algorithm from a new perspective," INFOCOM, 2011 Proceedings IEEE, April 2011.

Santos, A. F. et al. (2013), “Multi-gigabit traffic identification on GPU” In Proc. of the ACM HPPN '13 Workshop. ACM, NY, USA, 39-44.

Song, T. and Zhou, Z. (2013), "File-aware P2P traffic classification: An aid to network management", PPNA Journal (2013): 1-15.

Yu, F. et al. (2006), "Fast and memory-efficient regular expression matching for deep packet inspection," Architecture for Networking and Communications systems.

Zhanikeev, M. and Tanaka, Y., (2013) "A graphical method for detection of Flash Crowds in traffic." Telecommunication Systems (2013): 1-15.