

Contando os Segundos: Avaliação de Estratégias de Domínio Temporal para a Gerência de Regras em Redes SDN

Miguel Neves, Rodrigo Oliveira, Fabrício Mazzola, Daniel Marcon,
Luciano Gasparly, Marinho Barcellos

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{mcneves, ruas.oliveira, fmmazzola, dsmarcon, paschoal, marinho}@inf.ufrgs.br

Abstract. *Powered by Software Defined Networks (SDN), network administrators can achieve better control over traffic. This advantage comes at the cost of increased number of rules, which, in turn, leads to resource scarcity and lower performance. Recent proposals attempt to make better use of flow tables by optimizing the time each rule remains installed. In this paper, we perform an in-depth analysis of state-of-the-art strategies. First, we identify their principles and characterize their design space. Then, we perform an extensive experimental evaluation to show their benefits and limitations. Results indicate that the performance of current strategies is far from optimal and heavily relies on a precise parameterization. Our findings provide the foundations for the development of robust strategies to address the limitations of the literature.*

Resumo. *As Redes Definidas por Software (SDN) permitem aos administradores de rede um melhor controle sobre o tráfego de dados. Entretanto, tal vantagem requer um aumento do número de regras de encaminhamento a serem instaladas nas tabelas de fluxos de switches e roteadores, acarretando escassez de recursos e menor desempenho. Propostas recentes tentam fazer melhor uso das tabelas através da otimização do tempo que cada regra permanece instalada. Neste artigo, apresentamos uma avaliação qualitativa e quantitativa abrangente sobre as principais estratégias presentes no estado-da-arte. Primeiramente, identificamos os seus princípios e determinamos o espaço de projeto associado. Em seguida, realizamos extensas avaliações experimentais a fim de mostrar os seus benefícios e limitações. Resultados indicam que o desempenho dos métodos atuais está distante do comportamento ótimo e também depende fortemente de uma parametrização precisa. As conclusões obtidas servem de base para o desenvolvimento de estratégias mais robustas, que possam tratar as limitações encontradas na literatura.*

1. Introdução

A habilidade de gerenciar fluxos com granularidade fina é um dos principais benefícios das Redes Definidas por Software (SDN). Por exemplo, um provedor de infraestrutura de nuvem pode facilmente implementar mecanismos de reserva de banda entre pares de máquinas virtuais (VMs) ou um administrador de rede pode coletar estatísticas em nível de fluxo de forma eficiente utilizando dispositivos SDN [Moshref et al. 2013, Su et al. 2015]. Contudo, a possibilidade de gerenciar o tráfego da rede com tal granularidade possui custos associados, explicados a seguir.

Em redes SDN, políticas são expressas através de regras de encaminhamento, e instaladas em tabelas de fluxos nos dispositivos por meio de um controlador. Tais regras podem ter um tamanho considerável (até 44 campos de cabeçalho em versões recentes do protocolo OpenFlow [ONF 2015]), e ainda ocupar espaço de armazenamento e capacidade de processamento extras para a manutenção de contadores associados [Moshref et al. 2014]. Além disso, aliado à tendência crescente no tamanho das regras, os dispositivos SDN suportam uma quantidade baixa de entradas nas suas tabelas de fluxos, tipicamente uma ou duas ordens de grandeza menores do que o necessário [Katta et al. 2014]. Nesse contexto, o gerenciamento adequado das regras de encaminhamento é uma exigência.

O uso de mecanismos de gerência baseados em agregação (através de *wildcards*) ou posicionamento de regras, tipicamente empregados como solução, trazem a desvantagem de ofuscar informações importantes sobre os fluxos e aplicar desvios muitas vezes indesejados no caminho dos pacotes, respectivamente [Shirali-Shahreza and Ganjali 2015, Nguyen et al. 2015]. Por essa razão, propostas recentes tem buscado melhorar a eficiência no uso das tabelas de fluxos através de estratégias de gerência baseadas no tempo de duração das regras (ou seja, no tempo que as mesmas permanecem instaladas) [Zhang et al. 2014, Vishnoi et al. 2014, Kannan and Banerjee 2014]. Esse conjunto de propostas (definidas como sendo de domínio temporal), no entanto, carece de um estudo comparativo abrangente, capaz de delinear o espaço de projeto envolvido, determinar o desempenho relativo entre as estratégias e apontar direções relevantes de pesquisa.

Neste artigo, busca-se analisar em profundidade as principais propostas de domínio temporal para a gerência de regras e avaliar o impacto que as mesmas acarretam no desempenho de redes SDN. Os principais resultados obtidos indicam que: (i) as estratégias apresentam uma sobrecarga de utilização das tabelas de fluxos em torno de 15% em cenários de superprovisionamento de recursos; (ii) as estratégias são pelo menos 2x piores que o caso ótimo com relação ao número de intervenções do controlador em cenários de escassez de recursos; (iii) pequenas mudanças nas características do tráfego afetam consideravelmente o desempenho das estratégias; e (iv) o uso de mecanismos de timeout (sem um controle adequado) como ferramenta para a gerência de regras pode ser ruim na prática (ao contrário do que a literatura indica).

As principais contribuições deste artigo são: (i) o mapeamento do espaço de projeto e a análise das propriedades essenciais de cada estratégia; (ii) a avaliação comparativa do desempenho das estratégias considerando uma carga de trabalho representativa; e (iii) a identificação de direções relevantes de pesquisa, com base em resultados obtidos experimentalmente e de forma a proporcionar o desenvolvimento de novos estudos. Este é o primeiro trabalho a fazer uma análise sistemática e abrangente da área.

O restante do artigo está organizado da seguinte maneira. A Seção 2 introduz as principais propriedades que determinam o espaço de projeto das estratégias de gerenciamento temporal de regras, enquanto a Seção 3 apresenta modelos formais e discute as principais estratégias encontradas na literatura. A Seção 4, por sua vez, descreve a metodologia utilizada na avaliação de desempenho e os resultados obtidos. Finalmente, a Seção 5 discute trabalhos relacionados e a Seção 6 expõe as principais conclusões do estudo e perspectivas de trabalhos futuros.

2. Propriedades das Estratégias de Gerenciamento Temporal de Regras

As estratégias de gerenciamento temporal de regras podem ser caracterizadas conforme três propriedades centrais: a natureza do mecanismo de contagem da duração das regras; o grau de conhecimento que as estratégias detêm sobre o estado dos fluxos; e o algoritmo de cache aplicado para escolher uma regra a ser eliminada quando a tabela de fluxos se encontra cheia.

Natureza do tempo de duração (timeout) das regras. Grande parte das implementações de SDN suporta a eliminação automática de uma regra após o esgotamento do seu tempo de duração. Nesse sentido, as estratégias precisam considerar tanto o valor do timeout quanto o tipo de contador utilizado para controlá-lo, como explicado a seguir.

O valor do timeout pode ser definido *estaticamente* ou *dinamicamente*. No modo estático, um valor fixo é determinado previamente e utilizado em todas as regras instaladas. Esse valor não sofre variações e é aplicado mesmo se a duração dos fluxos mudar amplamente ao longo do tempo. Em contraste, no modo dinâmico, cada regra pode apresentar um valor diferente de timeout. Nesse caso, o timeout pode ser ajustado ao longo do tempo levando-se em conta o comportamento do fluxo associado (por exemplo, em relação ao tipo, tamanho e frequência de pacotes).

Existem dois tipos de contadores sobre a duração das regras: *por inatividade (idle)* e *rígido (hard)*. No modo inativo, a duração da regra é reajustada para o valor original sempre que acontece uma associação com pacotes. Em outras palavras, a entrada para o fluxo somente será removida se não acontecer nenhuma associação com pacotes durante a duração determinada para a regra. Em contraste, timeouts no modo rígido provocam a eliminação das regras caso as suas respectivas durações sejam transcorridas, independentemente das associações ocorridas.

Conhecimento sobre o estado dos fluxos. Algumas das características dos fluxos (incluindo duração, número de *bytes* e frequência de pacotes) podem afetar amplamente o desempenho dos mecanismos de timeout. Desse modo, um maior conhecimento sobre o estado de um dado fluxo é benéfico, pois decisões mais informadas podem ser feitas sobre a duração das regras. No entanto, a obtenção dessas informações pode ser custosa para a rede, uma vez que requer um processamento adicional nos nodos (switches e controlador) e normalmente gera uma troca extra de mensagens de controle. As estratégias costumam diferir amplamente em relação ao tipo e a quantidade de estado coletada da rede, o que é discutido na Seção 3.

Algoritmos de Cache de Regras. Tais algoritmos determinam qual será o comportamento da estratégia no momento em que a tabela estiver totalmente ocupada. Neste estado operacional, antes que uma nova regra seja instalada, o algoritmo de cache precisa eliminar uma entrada da tabela. Esses algoritmos possuem diferentes requisitos de processamento e sobrecargas, o que significa que a sua eficiência depende das condições operacionais da rede. Os algoritmos adotados pelas estratégias serão discutidos na Seção 3, onde as mesmas são formalmente modeladas e explicadas.

3. Modelagem das Estratégias

Esta seção visa modelar formalmente e discutir as principais estratégias encontradas na literatura, a fim de possibilitar uma comparação justa e precisa entre elas. Para auxiliar

nesse processo, a Tabela 1 resume os parâmetros usados em cada estratégia, enquanto a Tabela 2 mostra as suas informações mais relevantes.

Timeout Estático (ST). Essa estratégia representa o método padrão. A ideia principal consiste em encontrar um valor genérico (e estático) adequado para o tempo de duração de uma regra e aplicá-lo a todas as regras. Nesse caso, timeouts por inatividade são preferidos, pois evitam a eliminação de regras durante uma rajada de pacotes. Essa estratégia utiliza um único parâmetro (o valor estático do timeout - t_{ST}) e não possui nenhum algoritmo de cache associado. Assim como em [Vishnoi et al. 2014], uma eliminação aleatória foi considerada para situações nas quais a tabela está totalmente ocupada, mantendo a simplicidade da estratégia.

Remoção antecipada probabilística (probabilistic early eviction - PEV). Essa estratégia tenta eliminar regras inativas das tabelas de fluxos antes que os seus respectivos tempos de duração expirem. A eliminação é feita mesmo que a tabela não esteja cheia, visando remover entradas antes que os algoritmos de cache sejam acionados. Se ainda assim a tabela fica totalmente ocupada (por exemplo, devido à taxa de chegada de fluxos ser maior que a taxa de eliminação de regras), então o algoritmo LRU (*Least Recently Used*) é aplicado. Essa estratégia é composta por três ações: (i) determinação de um timeout - t_{PEV} (longo, estático e por inatividade) - para as regras; (ii) monitoramento do estado dos fluxos; e (iii) eliminação periódica (a cada DI unidades de tempo) de regras desnecessárias, a partir da probabilidade $P(i)$ do fluxo i ter encerrado.

Existem pelo menos duas maneiras diferentes de se determinar o melhor momento para a eliminação de uma regra (ou seja, $P(i)$). Em [Zarek 2012], um switch *OpenFlow* é estendido para procurar pacotes *SYN* e *FIN* em conexões TCP. Em [Kannan and Banerjee 2014], por sua vez, o controlador periodicamente tenta eliminar regras calculando $P(i)$ a partir das durações dos fluxos e dos intervalos entre pacotes monitorados na rede. Nosso modelo de PEV adota o método em [Kannan and Banerjee 2014], já que o mesmo independe de protocolos específicos, aplicando-se tanto a tráfego TCP como UDP.

Incremento Adaptativo (AI). Proposta por [Vishnoi et al. 2014], possui o objetivo de manter a utilização da tabela de fluxos baixa, administrando sob demanda fluxos longos. Essa estratégia é composta por dois passos: (i) estabelecer um timeout curto (t_{AI}^{min}) adequado para a maioria das regras; e (ii) atualizar a duração de uma regra a cada reinstalação até um valor máximo (t_{AI}^{max}), usando uma função de aumento $f(n_{reinstall})$. A função de aumento considera o número total de reinstalações ($n_{reinstall}$) de cada regra, sendo o controlador responsável por manter essa informação. Nosso modelo de AI, conforme sugerido pelos autores em [Vishnoi et al. 2014], adota um algoritmo aleatório de cache para eliminar regras quando a tabela de fluxos estiver cheia.

Timeout rígido adaptativo (Adaptive hard timeout - AHT). Proposta por [Zhang et al. 2014], essa estratégia determina dinamicamente uma duração única para todas as regras levando em conta características do tráfego, capacidades das tabelas de fluxos e restrições de desempenho da rede (por exemplo, tempo máximo de conclusão dos fluxos ou taxa máxima de intervenções do controlador). O objetivo é ajustar o tempo de vida das regras de acordo com as condições operacionais da infraestrutura. De forma geral, a estratégia é composta por três passos: (i) estimar a carga da rede como uma função

Parâmetro	Descrição
Timeout Estático (ST)	
t_{ST}	Timeout definido na instalação das regras
Remoção Antecipada Probabilística (PEV)	
t_{PEV}	Timeout definido na instalação das regras
DI	Intervalo periódico para tentar eliminar uma regra
$P(i)$	Probabilidade de eliminar uma regra inativa
Incremento Adaptativo (AI)	
t_{AI}^{min}	Timeout mínimo - Valor inicial para o timeout
t_{AI}^{max}	Timeout máximo - Limite superior para o timeout
$f(n_{reinstall})$	Função de incremento do timeout
Timeout Rígido Adaptativo (AHT)	
λ	Taxa estimada de chegada de fluxos
μ	Duração estimada dos fluxos
$C = c_1, c_2, \dots, c_n$	Conjunto de restrições de desempenho
$t_{AHT}(\lambda, \mu, R, [C])$	Timeout como uma função da carga da rede (λ, μ), do tamanho da tabela (R) e de restrições de desempenho ($[C]$)

Tabela 1. Lista de parâmetros usados em cada estratégia.

composta tanto pela taxa de chegada (λ) quanto pela duração (μ) dos fluxos; *(ii)* verificar se o valor atual estipulado para a duração das regras (t_{AHT}) satisfaz as restrições de desempenho (c_1, c_2, \dots, c_n) especificadas pelo administrador da rede; e *(iii)* se necessário, atualizar t_{AHT} de forma a atender todas as restrições.

A proposta de [Zhang et al. 2014] assume ausência de algoritmo de cache, em que regras serão removidas somente quando os seus respectivos tempos esgotarem. Essa premissa implica que novos fluxos deverão ser tratados diretamente pelo controlador até que alguma regra expire. Além disso, a mesma (premissa) pode ou não ser acompanhada por uma restrição de sobrecarga máxima do plano de controle (isto é, da taxa de intervenções do controlador). Caso seja, a duração das regras tende a ser gradualmente reduzida quando a tabela fica cheia, de forma a multiplexar o uso de suas entradas entre os fluxos ativos. Essa estratégia é baseada em timeouts rígidos, visto que tais timeouts refletem diretamente o tempo de duração dos fluxos.

Timeout ótimo (OPT). Para fins de comparação entre as estratégias, é descrita uma solução que minimiza o número de intervenções do controlador (ou seja, de mensagens de controle) na rede. Essa solução baseia-se na premissa de que informações precisas sobre o instante de chegada, o intervalo entre pacotes e a duração dos fluxos estariam disponíveis. Caso essas informações estivessem disponíveis, seria possível tratar o problema de determinar a duração das regras como um problema de escalonamento do uso da tabela de fluxos. Nesse sentido, uma regra pode ser removida da tabela assim que possível, reduzindo também o número de entradas ocupadas. A estratégia associada a essa solução opera da seguinte maneira: *(i)* realização da leitura de informações sobre os fluxos; e *(ii)* instalação de cada regra no momento em que o respectivo fluxo inicia, com duração idêntica à deste. Quando a tabela fica cheia, a regra referente ao fluxo cujo próximo pacote está mais distante no futuro é removida.

4. Avaliação

Nesta seção, apresentamos a avaliação realizada considerando diferentes cenários de operação da rede (p.ex. sobrecarga, tráfego variável). Entre os principais objetivos, está responder às seguintes questões: *(i)* qual a melhor estratégia de gerência temporal de regras para cada cenário? *(ii)* quais as melhores escolhas de projeto (por exemplo, favorecer timeouts longos ou curtos)?; e *(iii)* quão eficientes são as estratégias no geral (ou

Estratégia	Propostas	Natureza do Timeout	Algoritmo de cache	Conhecimento sobre o estado do fluxo	Descrição
ST	—	Estático / por inatividade	Aleatório	Baixo	Determinar timeout adequado e aplicá-lo para todas as regras
PEV	[Zarek 2012, Kannan and Banerjee 2014]	Estático / por inatividade	LRU	Alto	Usar timeout longo e periodicamente eliminar regras inativas
AI	[Vishnoi et al. 2014]	Dinâmico / por inatividade	Aleatório	Baixo	Aumentar o timeout quando a regra for reinstalada
AHT	[Zhang et al. 2014]	Dinâmico / rígido	—	Alto	Adaptar o valor do timeout baseado em características de tráfego, tamanho da tabela e desempenho esperado da rede
OPT	—	Dinâmico / rígido	Pacote mais distante no futuro	Alto	Possui, como premissa, conhecimento sobre o tempo de criação, finalização e chegada de pacotes para todos os fluxos

Tabela 2. Sumário de estratégias para gerenciamento temporal de regras.

seja, quão próximas estão do caso ótimo)?

O restante da seção está organizado como segue. Na Seção 4.1, a metodologia de avaliação (incluindo os fatores, as métricas, a carga de trabalho e o ambiente de testes usados nos experimentos) é apresentada. Em seguida, a Seção 4.2 descreve o processo de configuração dos parâmetros de cada estratégia. Por fim, na Seção 4.3 os resultados obtidos são discutidos.

4.1. Metodologia

Ambiente de testes. Os experimentos foram realizados usando-se Mininet HiFi (versão 2.1.0), POX (versão 0.1.0) e Open vSwitch (versão 1.9.0, com suporte a OpenFlow 1.0) para emular uma rede real. A rede emulada foi criada em uma máquina Intel i7 com 4 núcleos de 3.1 GHz e 16 GB de RAM. Nesse cenário, as estratégias oriundas da literatura (ST, PEV, AI e AHT) foram implementadas como aplicações para o controlador POX. O cenário ótimo (OPT), por sua vez, foi avaliado à parte através de simulações. Para facilitar a reprodução dos resultados, os códigos-fonte tanto das aplicações quanto do simulador utilizado encontram-se disponíveis online¹.

A topologia usada é constituída por dois hospedeiros (um cliente e um servidor) conectados por um único switch. Tal configuração permite isolar o comportamento das estratégias e ilustra de modo realista um cenário de gargalo de rede pelo qual grande número de fluxos necessita passar. Para desencadear a geração de regras de encaminhamento no controlador e o preenchimento da tabela de fluxos, foram criadas múltiplas conexões TCP, cada qual associada a um processo diferente nos hospedeiros. Nesse sentido, não houve instalação proativa de regras.

Carga de Trabalho. Para cada experimento, o cliente comunica-se com o servidor de acordo com dois parâmetros: duração do fluxo e intervalo de tempo entre rajadas de pacotes. A duração dos fluxos foi modelada como um processo log-normal com média $\mu = 4s$ e desvio padrão $\sigma = 1s$. O intervalo de tempo entre rajadas, por sua vez, segue um processo de Poisson com média $\lambda = 10s$. Tal configuração baseia-se em medições realizadas por trabalhos presentes na literatura [Benson et al. 2010, Karagiannis et al. 2004]. Em particular, a mesma reproduz um cenário realista onde a maioria dos fluxos tem curta duração e transmite pequenas quantidades de dados, enquanto alguns poucos fluxos (de duração mais longa) são responsáveis por boa parte do tráfego.

¹<https://github.com/mcnevesinf/SDN-timebased-rulemanagement>

A duração mínima dos fluxos e o menor intervalo de tempo entre rajadas foram definidos como 1s. Valores menores não alterariam os resultados obtidos, visto que esta é a granularidade máxima de tempo empregada em switches OpenFlow [ONF 2015]. O número de fluxos ativos na rede foi fixado em 300 (que se traduz em 600 regras considerando ambas as direções de uma conexão TCP), seguindo a variação no tamanho da tabela conforme descrito a seguir. Por fim, cada experimento teve duração de 10 minutos.

Tamanho da tabela. A capacidade da tabela de fluxos (em número de entradas) foi variada no intervalo [400,800]. De acordo com [Kuzniar et al. 2015, He et al. 2015], essa faixa de valores está em linha com dispositivos comerciais (de baixa capacidade). Esse intervalo foi escolhido, também, de maneira a evitar interferência decorrente de sobrecargas no ambiente de avaliação experimental. Observe que mais importante do que o número absoluto de entradas na tabela é a relação entre o seu tamanho e o número de fluxos. Variando o tamanho da tabela no intervalo escolhido, mediante uma carga fixa, é possível isolar o impacto de cada propriedade (timeout e algoritmo de cache) no desempenho das estratégias para diferentes estados operacionais da rede.

Métricas. O desempenho de cada estratégia foi avaliado segundo duas métricas, conforme a literatura [Zarek 2012]: número de intervenções do controlador e utilização da tabela de fluxos. Por intervenções do controlador, considerou-se tanto as operações de inserção de regras na tabela de fluxos (*flowmods*) quanto as ações de encaminhamento direto de pacotes (*packet outs*). O segundo caso é específico da estratégia AHT. A eliminação de regras (necessária na aplicação dos algoritmos de cache) foi desconsiderada, pois não influencia no desempenho dos fluxos [Bifulco and Matsiuk 2015]. O número de intervenções do controlador por fluxo é calculado pela adição entre o número de inserções (N_I) e o número de encaminhamentos diretos de pacotes (N_E), dividido pelo conjunto de todos os fluxos (F): $\frac{N_I + N_E}{|F|}$. Uma vez que cada intervenção aumenta substancialmente a latência do fluxo associado [Shirali-Shahreza and Ganjali 2015], valores menores são melhores.

A utilização da tabela de fluxos do switch, por sua vez, representa a porcentagem de entradas ocupadas (E_A) sobre o número total de entradas (E_T): $\frac{E_A}{E_T} * 100\%$. Para essa métrica, os valores devem ser os menores possíveis por duas razões: (i) a latência das operações de controle da tabela (i.e., inserção, remoção e modificação de entradas) aumenta de acordo com o número de regras instaladas [Kuzniar et al. 2015, He et al. 2015]; e (ii) quanto maior a utilização da tabela, maior a probabilidade que a mesma fique cheia (principalmente durante rajadas de fluxos) e, portanto, maiores as chances de intervenções do controlador serem necessárias [Guo et al. 2015].

4.2. Configurações Básicas

Para a realização de uma comparação justa, valores apropriados foram estabelecidos empiricamente para os parâmetros de cada estratégia. Esses valores foram definidos através de medições, considerando-se todas as combinações possíveis (avaliação fatorial completa) dos parâmetros listados na Tabela 3. A metodologia utilizada nesses experimentos preliminares também segue àquela descrita na seção anterior. Além disso, verificou-se uma tendência de piora nos resultados para a maioria dos parâmetros inteiros à medida que seus valores se aproximam dos limites testados, indicando que as faixas consideradas são representativas.

Parâmetros	Valores testados
Timeout Estático (ST)	
t_{ST} (em segundos)	5, 10, 20 , 30, 60
Remoção Antecipada Probabilística (PEV)	
t_{PEV} (em segundos)	5, 10, 20 , 30, 60
DI (em segundos)	5, 20, 60
$P(i)$	CDF da duração dos fluxos CDF do intervalo de tempo de chegada de pacotes
Incremento Adaptativo (AI)	
t_{AI}^{min} (em segundos)	1, 2, 5, 10, 20 , 30
t_{AI}^{max} (em segundos)	30, 60 , 120
$f(n_{reinstall})$	Linear: $t_{AI}^{min} \times n_{reinstall}$ Exponencial: $t_{AI}^{min} \times 2^{n_{reinstall}}$
Timeout Rígido Adaptativo (AHT)	
λ	De acordo com a carga de trabalho
μ	
$C = c_1, c_2, \dots, c_n$	Não foram consideradas restrições de desempenho da rede
$t_{AHT}(\lambda, \mu, R, [C])$ (em segundos)	10, 20, 30 (400-650) , 40 (700) , 50 (750) , 60 (800) 70, 80, 90, 100

Tabela 3. Resultado da análise de sensibilidade dos parâmetros.

A Tabela 3 também destaca as melhores configurações alcançadas para cada estratégia (valores em negrito). Note que a estratégia AHT apresentou configurações distintas para diferentes capacidades da tabela de fluxos (indicadas entre parênteses). Esse comportamento ocorre devido à estratégia considerar o tamanho da tabela no cálculo de duração das regras.

4.3. Resultados

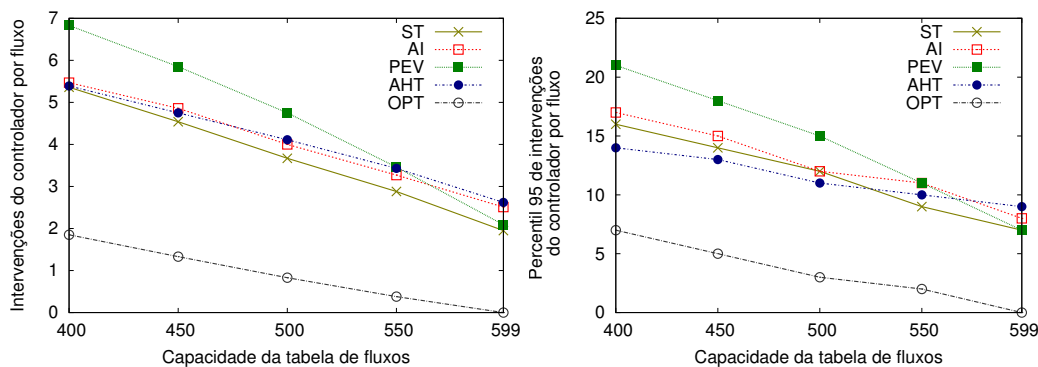
Os resultados estão organizados de acordo com três cenários distintos de operação da rede: escassez (Seção 4.3.1) e superprovisionamento (Seção 4.3.2) de recursos, ambos assumindo um tráfego estável, e situações de alta variabilidade nas características do tráfego (Seção 4.3.3). Um sumário com as principais observações pode ser encontrado na Seção 6.

4.3.1. Dispositivos de recursos limitados

Consideram-se dispositivos de recursos limitados os elementos de rede (switches e roteadores) cuja capacidade das tabelas de fluxos é menor que a demanda de regras durante a maior parte do período de operação da infraestrutura. Na prática, esses dispositivos normalmente envolvem os switches OpenFlow implementados em hardware com o uso de memória TCAM (*Ternary Content Addressable Memory*) e podem ser encontrados principalmente no núcleo das redes [Akyildiz et al. 2014].

Uma vez que a demanda (em número de regras) é maior que a capacidade das tabelas de fluxos, o componente dominante no desempenho das estratégias em cenários de recursos limitados é o seu algoritmo de cache. A Figura 1(a) mostra o número médio de intervenções do controlador por fluxo. Note que estratégias com diferentes algoritmos de cache apresentam desempenhos significativamente diferentes. Surpreendentemente, o algoritmo LRU teve um desempenho pior que o caso aleatório, ao contrário do que é indicado pela literatura [Zarek 2012].

A principal explicação é que o algoritmo LRU funciona bem quando o objetivo é diferenciar entradas ativas de entradas permanentemente inativas (ou com grande período de inatividade). Contudo, ele não tem um desempenho satisfatório quando todos os elementos do sistema (no caso, regras) encontram-se altamente ativos, e provavelmente serão



(a) Média de intervenções do controlador em um fluxo.

(b) Percentil 95 de intervenções do controlador em um fluxo.

Figura 1. Dispositivos de recursos limitados

utilizados ao menos mais uma vez. Nesse caso, espera-se que esquemas diretamente baseados no intervalo de tempo entre pacotes possam apresentar resultados melhores (vide estratégia OPT).

Conforme demonstrado pela Figura 1(a), à medida que a capacidade do dispositivo se aproxima da demanda (valores próximos de 600), mesmo estratégias com algoritmos de cache idênticos (por exemplo, ST e AI) tendem a apresentar desempenhos distintos. Esse comportamento evidencia o aumento da influência do mecanismo de timeout adotado pelas estratégias no seu desempenho. Note que todas as estratégias tiveram um desempenho pelo menos 2x pior que a estratégia ótima (OPT) para todos os cenários de baixa capacidade, mostrando que ainda há espaço para o desenvolvimento de algoritmos de cache mais eficientes.

Ademais, o número médio de intervenções por fluxo foi consideravelmente alto, sendo que grande parte dos fluxos (mais de 90% para uma tabela com capacidade de 400 entradas) teve a necessidade de pelo menos uma intervenção do controlador. O número de intervenções também variou bastante entre os fluxos (como mostrado na Figura 1(b)). De acordo com os resultados, pelo menos 5% dos fluxos apresentaram mais que o dobro da média de intervenções em todos os casos testados. Esse comportamento evidencia a necessidade da implantação de algum mecanismo de justiça entre os fluxos, principalmente em cenários onde os mesmos possuam algum tipo de *deadline* associado [Zhang et al. 2015].

4.3.2. Dispositivos de grande capacidade

Em contraste aos dispositivos de recursos limitados, os dispositivos de grande capacidade representam os elementos de rede cuja capacidade da tabela de fluxos é igual ou superior à demanda de regras durante grande parte do período de operação da rede. Na prática, esse grupo de dispositivos normalmente compreende os switches OpenFlow implementados em hardware com o uso de memória RAM e os software switches (por exemplo, Open vSwitch [Pfaff et al. 2015]). Tais dispositivos tendem a possuir taxas menores de processamento de pacotes que os dispositivos de recursos limitados e, por essa razão, geralmente são utilizados na borda das redes (por exemplo, junto aos hipervisores em redes

de *data center*) [Akyildiz et al. 2014].

Nos casos de grande capacidade, o componente determinante no desempenho das estratégias tende a ser o seu mecanismo de timeout, uma vez que a tabela tem menos chances de ficar cheia e, por consequência, os algoritmos de cache são pouco acionados. A exceção são os cenários de timeout excessivamente longos, que tendem a manter regras inativas nas tabelas.

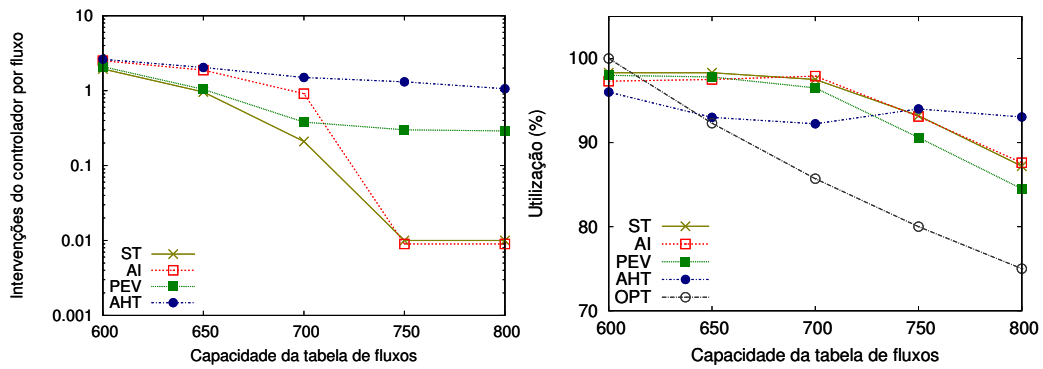
A Figura 2(a) mostra o número médio de intervenções do controlador por fluxo para cenários com dispositivos de grande capacidade, estando o eixo y representado em escala logarítmica. Note que a estratégia OPT está ausente porque não há necessidade de intervenções do controlador para o caso ótimo, uma vez que é possível manter cada regra instalada durante a duração exata do respectivo fluxo. Com relação às demais estratégias, a maioria apresenta uma quantidade baixa de intervenções (menos de 1 por fluxo na média), com um percentual significativo dos fluxos não necessitando de intervenção alguma. A principal conclusão é que, com uma parametrização adequada, é possível atingir um alto grau de eficiência no gerenciamento temporal das regras, mesmo quando estratégias simples (como o uso isolado de timeout estático) são consideradas (vide ST). Afinal, existe sempre um tamanho de tabela suficientemente grande para comportar todas as regras e evitar a necessidade de uma nova intervenção do controlador.

A Figura 2(b), por sua vez, mostra a utilização média da tabela de fluxos para cada experimento realizado. A maioria das estratégias tende a utilizar entre 10 e 15% a mais os recursos em comparação ao caso ótimo (OPT). De maneira geral, a utilização relaciona-se inversamente e de forma não-linear com o número de intervenções do controlador (ou seja, uma pequena redução no uso da tabela pode implicar em um aumento significativo da necessidade de intervenções). Considerando as estratégias analisadas, a tentativa de retirada antecipada de uma regra tende a amplificar mais de 10x (vide PEV) a necessidade de intervenções do controlador. O mesmo efeito se aplica ao uso de timeouts excessivamente curtos. Note que a estratégia OPT usa inicialmente 100% da capacidade da tabela, uma vez que ela não tenta remover regras de fluxos que ainda podem estar ativos havendo espaço suficiente.

Considerando que a latência de comunicação com o controlador é tipicamente uma a duas ordens de grandeza maior que a latência de uma operação de controle sobre a tabela (por exemplo, inserção ou modificação de uma regra) [Bifulco and Matsiuk 2015], estratégias que priorizam a redução no número de intervenções em detrimento da utilização (por exemplo, ST e AI) são preferíveis. Por fim, estratégias baseadas em mecanismos rígidos de timeout tendem a apresentar tanto índices de utilização quanto de intervenções elevados, mesmo considerando características da carga na sua definição (vide AHT). Provavelmente esse efeito é decorrente da granularidade (baixa) desse tipo de timeout, que requer escalas maiores de valores.

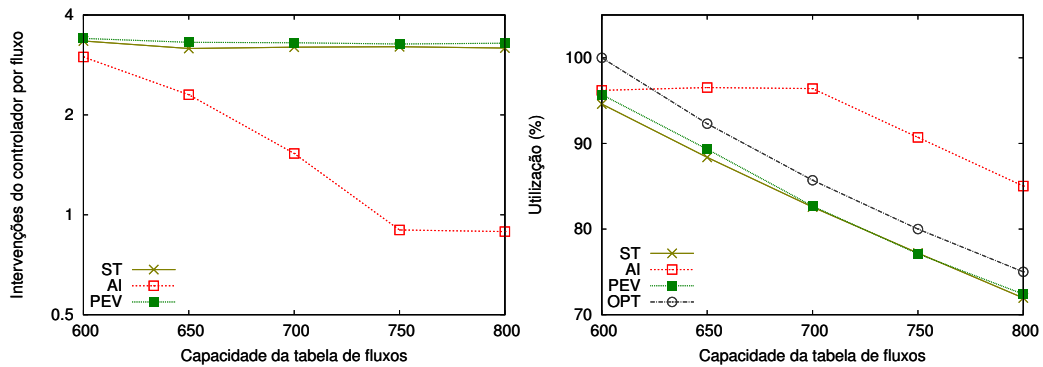
4.3.3. Cenários de parametrização imprecisa

Diversos estudos apontam variabilidades nas características do tráfego das redes quando longos períodos (isto é, ordens de grandeza de segundos ou maiores) são considerados [Cordero and Bonaventure 2014, Almeida et al. 2015, Duarte et al. 2014]. Nesse sentido,



(a) Média de intervenções do controlador em um fluxo.

(b) Utilização média da tabela de fluxos.

Figura 2. Dispositivos de grande capacidade.

(a) Média de intervenções do controlador em um fluxo.

(b) Utilização média da tabela de fluxos.

Figura 3. Cenários de parametrização imprecisa.

imprecisões podem ocorrer com relação à parametrização das estratégias. Nesta seção, os principais resultados obtidos para cenários imprecisos são discutidos.

De forma geral, os parâmetros podem estar subestimados ou superestimados, resultando na ocorrência de regras com durações excessivamente curtas ou longas, respectivamente. Nesse contexto, a Figura 3 mostra o número de intervenções do controlador e a utilização da tabela de fluxos quando os parâmetros estão subestimados. Para esses cenários, adotou-se a configuração de parâmetros que levou aos resultados mais próximos daqueles obtidos na Seção 4.2 para o melhor caso. Em outras palavras, $t_{ST} = t_{PEV} = t_{AI}^{min} = 10s$. Os demais parâmetros não foram alterados. Finalmente, a estratégia AHT não foi considerada, uma vez que ela leva em conta a carga da rede na determinação da duração das regras.

Pode-se perceber que pequenas mudanças na qualidade da parametrização das estratégias (ou, de forma equivalente, nas características do tráfego) podem ocasionar grandes variações de desempenho (Figuras 2 e 3). Nesse contexto, estratégias envolvendo o uso de timeouts dinâmicos tendem a apresentar melhores resultados, uma vez que é possível adaptá-las à carga da rede de modo mais eficiente. O processo de adaptação, no entanto, também tem um custo não negligível, percebido quando comparamos as Figuras 2(a) e 3(a) (vide estratégia AI) e os índices de utilização da tabela de fluxos (Figura

3(b)). Note que o algoritmo que minimiza o número de intervenções (OPT) pode levar, por outro lado, a níveis de ocupação da tabela maiores. Ademais, quando parâmetros superestimados são considerados, o mesmo comportamento geral se repete.

5. Trabalhos Relacionados

Além do grupo de estratégias estudado neste artigo, a literatura sobre gerenciamento de regras em redes SDN compreende propostas que buscam reduzir a quantidade total de regras necessárias para a implantação de uma determinada política. Nesse sentido, [Voellmy et al. 2013] e [Schlesinger et al. 2014] tentam otimizar o processo de transformação de políticas de alto nível nas suas respectivas regras através do uso de *frameworks* (isto é, linguagens e compiladores) específicos para a expressão dessas políticas. [Kang et al. 2013] e [Moshref et al. 2013], por sua vez, atuam posicionando estrategicamente as regras na rede de forma a permitir que cada regra possa ser aplicada ao maior número possível de fluxos, bem como a minimizar a sobrecarga de tráfego gerada com prováveis desvios nos caminhos dos pacotes. Tais propostas são ortogonais às avaliadas neste trabalho, podendo ambos os grupos serem implantados de forma conjunta e independente em um ambiente de rede SDN.

6. Conclusão

Neste artigo, analisamos o espaço de projeto e comparamos o desempenho de diferentes estratégias de domínio temporal para o gerenciamento de regras em redes SDN. Este é o primeiro trabalho a fazer um estudo sistemático e abrangente da área. As observações-chave obtidas foram as seguintes:

- Conforme esperado, quando há um superprovisionamento de recursos na rede, a maioria das estratégias apresenta uma quantidade baixa de intervenções do controlador (menor que uma por fluxo na média), com um percentual significativo dos fluxos não necessitando de intervenção alguma. Em contrapartida, elas tendem a ter um nível de utilização entre 10 e 15% maior de recursos em comparação com a estratégia OPT.
- A utilização da tabela relaciona-se de forma inversa e *não-linear* com o número de intervenções do controlador (que cresce rapidamente). Dessa forma, regras de duração longa são preferíveis em cenários de incerteza.
- Pequenas mudanças nas características do tráfego podem afetar consideravelmente o desempenho das estratégias. Assim, estratégias envolvendo o uso de *time-outs* dinâmicos tendem a apresentar melhores resultados, pois podem se adaptar às variações da carga. Contudo, mesmo o processo de adaptação possui um custo não negligível.
- Em cenários de escassez de recursos, o desempenho das estratégias varia significativamente de acordo com o algoritmo de cache utilizado. No geral, todas as estratégias foram pelo menos 2x piores que o caso ótimo com relação ao número de intervenções do controlador. Além disso, houve uma ampla variação do número de intervenções entre os fluxos.

Acreditamos que tais observações poderão qualificar o desenvolvimento de novos estudos e alimentar avanços científicos na área, fundamental para o uso eficiente dos recursos de encaminhamento da rede. Como trabalhos futuros, pretende-se estender

a avaliação realizada para considerar outros fatores, tais como múltiplas tabelas de fluxos, diferentes arquiteturas de cache e cenários mais complexos de rede, que envolvam agregação de regras (regras com wildcards), topologias maiores e dispositivos de encaminhamento heterogêneos. Também pretende-se investigar mecanismos para melhorar o desempenho das estratégias estudadas, potencialmente adicionando-se formas de oferecer garantias de desempenho aos fluxos (por exemplo, tempo máximo de conclusão).

Agradecimentos. Essa pesquisa é financiada em parte pelos projetos Phoenix (MCTI/CNPq 14/2014) e ProSeG (MCTI/CNPq/CT-ENERG 33/2013).

Referências

- Akyildiz, I. F., Lee, A., Wang, P., Luo, M., and Chou, W. (2014). A roadmap for traffic engineering in sdn-openflow networks. *Computer Networks*, 71:1 – 30.
- Almeida, W., Santos, B., Vieira, A., Cunha, I., and Almeida, J. (2015). Caracterização da transmissão de um grande evento esportivo. In *33o. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2015*.
- Benson, T., Akella, A., and Maltz, D. A. (2010). Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement - IMC'10*.
- Bifulco, R. and Matusiuk, A. (2015). Towards scalable sdn switches: Enabling faster flow table entries installation. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM'15*.
- Cordero, J. and Bonaventure, O. (2014). Understanding the topological properties of internet traffic: A view from the edge. In *2014 IFIP Networking Conference*.
- Duarte, R., Cunha, I., Almeida, J., and Vieira, A. (2014). Avaliação do impacto de falhas na rede nacional de ensino e pesquisa no tráfego de um campus universitário. In *32o. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2014*.
- Guo, Z., Xu, Y., Cello, M., Zhang, J., Wang, Z., Liu, M., and Chao, H. J. (2015). Jump-flow: Reducing flow table usage in software-defined networks. *Computer Networks*, 92:300 – 315.
- He, K., Khalid, J., Gember-Jacobson, A., Das, S., Prakash, C., Akella, A., Li, L. E., and Thottan, M. (2015). Measuring control plane latency in sdn-enabled switches. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research - SOSR'15*.
- Kang, N., Liu, Z., Rexford, J., and Walker, D. (2013). Optimizing the "one big switch" abstraction in software-defined networks. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies - CoNEXT'13*.
- Kannan, K. and Banerjee, S. (2014). Flowmaster: Early eviction of dead flow on sdn switches. In *LNCS, Distributed Computing and Networking*, volume 8314, pages 484 – 498.
- Karagiannis, T., Molle, M., Faloutsos, M., and Broido, A. (2004). A nonstationary poisson view of internet traffic. In *2004 IEEE Conference on Computer Communications - INFOCOM'04*.

- Katta, N., Alipourfard, O., Rexford, J., and Walker, D. (2014). Infinite cache flow in software-defined networks. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking - HotSDN'14*.
- Kuzniar, M., Peresini, P., and Kostic, D. (2015). What you need to know about sdn flow tables. In *LNCS, Passive and Active Measurement*, volume 8995, pages 347 – 359.
- Moshref, M., Yu, M., Govindan, R., and Vahdat, A. (2014). Dream: Dynamic resource allocation for software-defined measurement. In *Proceedings of the 2014 ACM Conference on Special Interest Group on Data Communication - SIGCOMM'14*.
- Moshref, M., Yu, M., Sharma, A., and Govindan, R. (2013). Scalable rule management for data centers. In *10th USENIX Symposium on Networked Systems Design and Implementation - NSDI'13*.
- Nguyen, X.-N., Saucez, D., Barakat, C., and Turletti, T. (2015). Officer: A general optimization framework for openflow rule allocation and endpoint policy enforcement. In *2015 IEEE Conference on Computer Communications - INFOCOM'15*.
- ONF (2015). Openflow switch specification. version 1.5.1 (protocol version 0x06). Disponível em: <https://www.opennetworking.org>. Acesso em Novembro, 2015.
- Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., and Casado, M. (2015). The design and implementation of open vswitch. In *12th USENIX Symposium on Networked Systems Design and Implementation - NSDI'15*.
- Schlesinger, C., Greenberg, M., and Walker, D. (2014). Concurrent netcore: From policies to pipelines. In *Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming - ICFP'14*.
- Shirali-Shahreza, S. and Ganjali, Y. (2015). Rewiflow: Restricted wildcard openflow rules. *SIGCOMM Comput. Commun. Rev.*, 45(5):29–35.
- Su, Z., Wang, T., Xia, Y., and Hamdi, M. (2015). Cemon: A cost-effective flow monitoring system in software defined networks. *Computer Networks*, 92:101 – 115.
- Vishnoi, A., Poddar, R., Mann, V., and Bhattacharya, S. (2014). Effective switch memory management in openflow networks. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems - DEBS'14*.
- Voellmy, A., Wang, J., Yang, Y. R., Ford, B., and Hudak, P. (2013). Maple: Simplifying sdn programming using algorithmic policies. In *Proceedings of the 2013 ACM Conference on Special Interest Group on Data Communication - SIGCOMM'13*.
- Zarek, A. (2012). OpenFlow Timeouts Demystified. Master's thesis, University of Toronto, Canada.
- Zhang, H., Shi, X., Yin, X., Ren, F., and Wang, Z. (2015). More load, more differentiation — a design principle for deadline-aware congestion control. In *2015 IEEE Conference on Computer Communications - INFOCOM'15*.
- Zhang, L., Lin, R., Xu, S., and Wang, S. (2014). AHTM: Achieving efficient flow table utilization in software defined networks. In *2014 IEEE Global Communications Conference - GLOBECOM'14*.