

Temporizadores Dinâmicos no Plano de Dados P4 para Mitigação de Ataques Furtivos de Exaustão de TCAM

Ryan Borges de Leon¹, Bruno L. Dalmazo¹

¹Centro de Ciências Computacionais (C3)
Universidade Federal do Rio Grande (FURG) Rio Grande — RS — Brasil

ryanbleon0@furg.br, dalmazo@furg.br

Abstract. *Software-Defined Networking (SDN) separates control and data planes, both finite resources prone to contention. Low-rate flow-table exhaustion attacks overflow the switch table while saturating the controller, evading volumetric detectors. Dynamic-timeout defenses are mature on OpenFlow, with controller-side decisions; on the P4 programmable data plane, defenses against low-rate exhaustion remain scarce and do not exploit dynamic timeouts. This paper systematizes related work across four dimensions (plane, attack profile, defense mechanism, decision locus) and identifies an unaddressed gap: P4, stealthy exhaustion, and in-data-plane dynamic timeout. We propose an architecture in which flow-table management and timeout enforcement run entirely in the data plane, and outline a grounded evaluation methodology.*

Resumo. *Redes Definidas por Software (SDN) separam os planos de controle e de dados, ambos recursos finitos e sujeitos a contenção. Ataques de exaustão de tabela de fluxo de baixa taxa transbordam a tabela do switch enquanto saturam o controlador, evadindo detectores volumétricos. Defesas por timeout dinâmico são maduras em OpenFlow, com decisão no controlador; no plano de dados programável P4, as defesas contra exaustão de baixa taxa ainda são poucas e não exploram a temporização dinâmica. Este artigo sistematiza a literatura relacionada em quatro dimensões (plano, perfil de ataque, mecanismo de defesa e local da decisão) e identifica uma lacuna não abordada: P4, exaustão furtiva e timeout dinâmico no plano de dados. Propomos uma arquitetura em que a gestão da tabela e a aplicação do timeout ocorrem inteiramente no plano de dados e delineamos uma metodologia de avaliação fundamentada.*

1. Introdução

Redes Definidas por Software (*Software-Defined Networking* – SDN) desacoplam a lógica de controle, centralizada no controlador, do encaminhamento nos switches. Esse desenho melhora a programabilidade, mas transforma dois elementos em recursos finitos e compartilhados: a tabela de fluxo, implementada em memória TCAM cara e limitada [Ertekin et al. 2025], e o canal de controle, ambos alvos diretos de ataques de negação de serviço.

Diferentemente dos volumétricos, os ataques furtivos de exaustão operam em baixa taxa: um fluxo contínuo de pacotes com origens forjadas (*spray-and-forget*) força a instalação incessante de entradas, transbordando a tabela e, a cada fluxo novo, disparando

um evento *packet-in/digest* para o controlador. Esses ataques têm duas propriedades perigosas: a taxa mínima para transbordar a tabela é pequena [Cao et al. 2017], o que os torna invisíveis a defesas baseadas em volume [Pascoal et al. 2020]; e o próprio tratamento de fluxos novos amplifica a carga sobre o controlador [Jain et al. 2025], tornando o plano de controle (não só a tabela) o recurso exaurido. Novos ataques [Wang et al. 2025] e defesas [Tang et al. 2024, Jain et al. 2025] seguem surgindo.

A defesa por temporização dinâmica, que ajusta o *timeout* das entradas conforme a pressão observada e as expurga mais cedo sob ataque, é uma resposta natural e já validada [Isyaku et al. 2020, Tang et al. 2024, Huang et al. 2024]. Esse respaldo, porém, restringe-se a implementações OpenFlow, nas quais a decisão de mitigação reside no controlador. No plano de dados programável P4, as defesas contra exaustão de baixa taxa ainda são poucas: a identificada nesta revisão classifica o ataque e bloqueia a origem [Jain et al. 2025], sem ajuste de temporização.

Este é um artigo de posicionamento (trabalho em andamento). Suas contribuições são: (i) uma sistematização das defesas de tabela de fluxo segundo os eixos plano, perfil de ataque, mecanismo e local da decisão (Seção 3, Tabela 1); (ii) a delimitação da lacuna acima; (iii) uma arquitetura de mitigação em P4/BMv2 com gestão da tabela e do *timeout* no plano de dados (Seção 4); e (iv) uma metodologia de avaliação fundamentada em caracterizações empíricas de tráfego.

2. Fundamentação Teórica

Esta seção apresenta o plano de dados P4, o limite das tabelas de fluxo e os ataques que o exploram.

2.1. Plano de dados programável P4 e a oportunidade

A linguagem P4 permite programar o switch no próprio plano de dados, abrindo uma possibilidade ausente em OpenFlow: mover a lógica de mitigação para dentro do switch, evitando o *round-trip* por fluxo novo que transforma o controlador em vetor de amplificação. P4LRU [Zhao et al. 2023] demonstrou a viabilidade de manter estruturas de estado sofisticadas inteiramente no plano de dados, com uma política de substituição LRU quase ótima sem envolver o controlador. A hipótese deste trabalho é que o mesmo vale para a temporização dinâmica sensível à pressão de ataque.

2.2. Tabela de fluxo e o limite de TCAM

As regras de encaminhamento são instaladas em tabelas *match-action* sobre TCAM, restrita a poucos milhares de entradas por custo e consumo de energia [Yu et al. 2021]. Esgotada a tabela, o switch rejeita a inserção de novas entradas (erro *table-full*); o expurgo automático existe apenas como recurso opcional, desabilitado por padrão [Open Networking Foundation 2015], e, sem ele, o destino de um pacote sem regra correspondente depende da política de *table-miss*: descartado ou reenviado repetidamente ao controlador sem nunca obter regra própria, ampliando a carga de controle. O expurgo antecipado mitiga o cenário, reduzindo atraso e elevando vazão em relação à expiração apenas por *timeout*, mas expurgar cedo demais devolve ao controlador pacotes de fluxos ainda ativos [Humayun et al. 2021]. Internalizar a gestão da tabela no plano de dados visa eliminar esse retorno, e calibrar a agressividade do expurgo é o compromisso investigado neste trabalho (Seção 4).

2.3. Ataques de exaustão de baixa taxa

O ataque LOFT [Cao et al. 2017] formaliza a taxa mínima que transborda a tabela (Figura 1), estabelecendo o ataque furtivo como classe distinta da volumétrica e evasiva a defesas de volume [Pascoal et al. 2020]. Modelos inteligentes exploram o comportamento de cache da tabela para maximizar o dano com esforço mínimo [Yu et al. 2021]. A origem forjada é parte do vetor: cada pacote com IP de origem falsificado é tratado pelo switch como um fluxo novo [Dai et al. 2023], e a falsificação de IP e MAC é empregada inclusive na variante de baixa taxa [Himanshu et al. 2024].

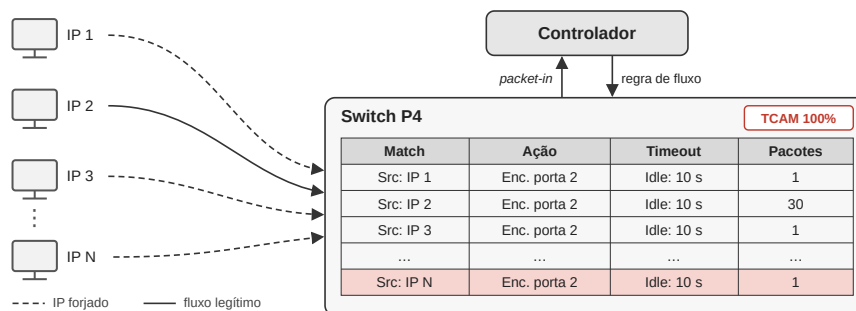


Figura 1. Exaustão furtiva: IPs forjados criam fluxos novos que saturam a TCAM.

Fonte: Elaborado pelos autores.

3. Trabalhos Relacionados e Delimitação da Lacuna

Esta seção descreve a revisão da literatura e a sistematização que delimita a lacuna.

3.1. Protocolo e escopo da revisão

A busca automática cobriu IEEE Xplore, ACM DL e ScienceDirect (2020–2026) com a *string* ("SDN" OR "P4" OR "Programmable Switch") AND ("dynamic timeout" OR "adaptive timeout" OR "early eviction" OR "flow table management" OR "TCAM exhaustion"). Incluíram-se estudos que tratam do esgotamento de recursos do switch ou da proteção do controlador contra ataques indiretos (CI1), publicados entre 2020 e 2026 (CI2) e que aplicam alguma técnica de gerência de tabela (CI3); excluíram-se os restritos a DDoS sem abordar *timeout* ou esgotamento de recursos (CE1), os voltados a *Machine Learning* no controlador (CE2) e os sem abordagem de prevenção (CE3).

A busca retornou 159 estudos (40 no IEEE Xplore, 51 no ACM DL e 68 no ScienceDirect), dos quais 98 publicados entre 2020 e 2026; após a triagem, restaram **20 estudos primários** (7, 7 e 6 por base, respectivamente).

Referências metodológicas (Seções 4 e 5) e trabalhos próximos não capturados pela triagem por título (Isyaku et al. [Isyaku et al. 2020] e FTODefender [Tang et al. 2024]) foram obtidos por *snowballing*.

3.2. Sistematização e a lacuna

A literatura pode ser lida em três blocos: o **ataque** (LOFT [Cao et al. 2017], *slow-rate* [Pascoal et al. 2020], inferência de cache [Yu et al. 2021], todos em OpenFlow); a **defesa por gerência de tabela**, onde temporização dinâmica é técnica madura (Isyaku et al. [Isyaku et al. 2020], FTODefender [Tang et al. 2024], *timeout hopping* [Huang et al. 2024], que também varia o *timeout* contra exaustão de baixa taxa), mas,

quando aplicada contra ataques, sempre em OpenFlow e com decisão no controlador; e o **plano de dados P4**, com foco em estruturas de estado (P4LRU [Zhao et al. 2023]) ou em outras classes de ataque [Avinash Reddy et al. 2025]. Nele, as defesas contra exaustão de baixa taxa ainda são poucas: a identificada nesta revisão, FTSheild [Jain et al. 2025], classifica o ataque e bloqueia a origem, estratégia não avaliada com IP forjado e que, diante dele, tende a perder efeito, já que cada pacote forjado constitui um fluxo novo (Seção 2.3). Nenhuma das defesas identificadas ajusta o *timeout* das entradas no plano de dados, atuando sobre o recurso (o tempo de vida das entradas) sem depender de identificar o atacante.

A Tabela 1 sistematiza esses trabalhos; os demais estudos da RSL tratam de eficiência de tabela sem modelo de ataque. Lendo-a por colunas, nenhum trabalho reúne simultaneamente as quatro células que definem esta proposta: plano P4, ataque furtivo, temporização dinâmica e decisão no plano de dados.

Tabela 1. Sistematização dos trabalhos relacionados (•: contempla a dimensão; †: obtido por *snowballing*, os demais pertencem à RSL).

	Dimensão	LOFT [†]	Slow-DoS	FT-Sec	FTOA-RP	FTODef. [†]	Isyaku [†]	T-Hop	EarlyEv	ATEO	Scotch	RAPID	P4LRU	P4-FTMod [†]	FTSheild	Este trab.
Plano	P4 / plano prog. OpenFlow / OVS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Ataque	Volumétrico										•	•				
	Furtivo / baixa taxa	•	•	•	•	•		•				•		•	•	•
	Timeout dinâmico					•	•	•		•						•
Defesa	Expurgo / substituição					•	•		•	•			•			
	Deteção											•		•	•	
Decisão	Controlador					•	•	•		•	•	•			•	
	Plano de dados												•	•	•	•
Foco	Exaustão de tabela	•	•	•	•	•	•	•	•	•		•	•	•	•	•
	Exaustão do control.		•								•	•			•	•

Referências: LOFT [Cao et al. 2017]; Slow-DoS [Pascoal et al. 2020]; FT-Sec [Yu et al. 2021]; FTOA-RP [Wang et al. 2025]; FTODef. [Tang et al. 2024]; Isyaku [Isyaku et al. 2020]; T-Hop [Huang et al. 2024]; EarlyEv [Humayun et al. 2021]; ATEO [Ertekin et al. 2025]; Scotch [Dai et al. 2023]; RAPID [Himanshu et al. 2024]; P4LRU [Zhao et al. 2023]; P4-FTMod [Avinash Reddy et al. 2025]; FTSheild [Jain et al. 2025].

4. Proposta e Metodologia

Esta seção apresenta a arquitetura proposta e a metodologia de avaliação planejada.

4.1. Arquitetura

A proposta é um mecanismo de **temporização dinâmica sensível à pressão de ataque**: o *timeout* das entradas é longo em condições saudáveis e reduzido sob indício de exaustão (expurgo antecipado), limitando a ocupação da tabela. Como no P4LRU [Zhao et al. 2023], a tabela de fluxo é mantida em registradores do switch, indexados por *hash*, e gerida no plano de dados; o critério de remoção, porém, é temporal: cada entrada guarda o instante do seu último pacote e expira quando um pacote a encontra vencida ou quando uma varredura incremental a alcança. Como P4 não dispõe de exponencial, os valores de τ serão pré-calculados e carregados uma única vez pelo controlador em uma tabela indexada pela ocupação; fora isso, o controlador apenas instala as rotas, sem decisões por fluxo.

O Algoritmo 1 formaliza esse mecanismo em dois regimes, a partir da ocupação atual (ρ) e da taxa de chegada de fluxos (Δ). O decaimento exponencial do *timeout* em função da ocupação segue o ATEO [Ertekin et al. 2025], proposto para OpenFlow com decisão no controlador; este trabalho acrescenta a projeção da ocupação ($\hat{\rho}$), que antecipa a reação à saturação, a recuperação linear quando a pressão cede e a execução no plano de dados. Os parâmetros (ρ_{safe} , ρ_{crit} , λ , β) permanecem simbólicos: a calibração virá das características de tráfego discutidas a seguir.

Algoritmo 1 Temp. Dinâmica por Controle Preditivo Exponencial-Linear

```

1: procedure TIMEOUTDINAMICO( $\rho, \Delta, \tau_{\text{base}}$ )
2:    $\hat{\rho} \leftarrow \rho + \Delta \cdot t_{\text{lookahead}}$ 
3:   if  $\hat{\rho} > \rho_{\text{crit}}$  then
4:      $\tau \leftarrow \tau_{\text{base}} \cdot \exp(-\lambda \cdot (\hat{\rho} - \rho_{\text{safe}}))$ 
5:   else
6:      $\tau \leftarrow \tau_{\text{base}} \cdot \left(1 + \beta \cdot \frac{\rho_{\text{safe}} - \rho}{\rho_{\text{safe}}}\right)$ 
7:   end if
8:    $\tau \leftarrow \max(\tau_{\text{min}}, \min(\tau_{\text{max}}, \tau))$ 
9:   return  $\tau$ 
10: end procedure

```

4.2. Ambiente, topologia e tráfego

A avaliação planejada, ainda não executada, usará Mininet com o *target* BMv2 (*simple_switch_grpc*) em topologia *leaf-spine* reduzida (dois *leaf*, um *spine*), que concentra o tráfego leste-oeste no *spine*, ponto de contenção conhecido *a priori*. A capacidade da tabela é reduzida no programa P4 para acelerar a saturação nos testes. O tráfego legítimo seguirá distribuições empíricas de tamanho de fluxo em *datacenter* (*Web Search* [Alizadeh et al. 2010] e *Data Mining* [Greenberg et al. 2009]) e o malicioso, o dataset HLD-DDoSSDN [Bahashwan et al. 2024]; datasets DDoS convencionais foram descartados por não capturarem fenômenos do plano de controle. Os limiares de decisão serão derivados dessas características, não de valores *ad hoc*. Duas questões orientam o trabalho: (RQ1) temporização dinâmica no plano de dados limita a ocupação da tabela sob exaustão furtiva sem envolver o controlador? (RQ2) qual o compromisso entre agressividade do expurgo, dano colateral a fluxos legítimos e parametrização do *timeout* a partir de tráfego real?

5. Considerações Finais e Trabalhos Futuros

Nenhum dos trabalhos identificados combina temporização dinâmica, plano de dados P4 e ataques furtivos de exaustão. Para preencher essa lacuna, propusemos uma arquitetura e um plano de avaliação; nesta etapa, a contribuição consiste na delimitação do problema e na definição dos componentes da solução.

Dois eixos guiarão a continuidade: quantificar o dano colateral identificado na RQ2 (entradas legítimas expulsas cedo demais precisam ser reinseridas e, com a tabela pressionada, podem ser recusadas, custo que em OpenFlow recai sobre o controlador); e investigar *Generative Adversarial Networks* (GANs) para gerar tráfego furtivo de alta fidelidade, exigindo síntese posterior de pacotes e grande volume de dados de treino.

Referências

Alizadeh, M. et al. (2010). Data center TCP (DCTCP). In *ACM SIGCOMM*.

- Avinash Reddy, B., Sagar Sahoo, K., and Bhuyan, M. (2025). Toward mitigation of flow table modification attacks in P4-based SDN data plane. *Security and Privacy*, 8:e70008.
- Bahashwan, A. A. et al. (2024). HLD-DDoSSDN: High and low-rates dataset-based DDoS attacks against SDN. *PLOS ONE*.
- Cao, J., Xu, M., Li, Q., Sun, K., Yang, Y., and Zheng, J. (2017). Disrupting SDN via the data plane: A low-rate flow table overflow attack. In *SecureComm*.
- Dai, Y., Wang, A., Guo, Y., and Chen, S. (2023). Elastically augmenting the control-path throughput in SDN to deal with internet DDoS attacks. *ACM Trans. Internet Technol.*, 23(1):9:1–9:25.
- Ertekin, E., Yiğit, B., and Alagöz, F. (2025). ATEO: Adaptive timeout & eviction orchestrator. In *IEEE BlackSeaCom*.
- Greenberg, A. et al. (2009). VL2: A scalable and flexible data center network. In *ACM SIGCOMM*.
- Himanshu, Saha, K., Das, P., and De, S. (2024). A network segmentation architecture for flow aggregation and DDoS mitigation in SDN using RAPID flow rules. In *ICDCN*, pages 232–241.
- Huang, C., Zhao, C., and Liao, L. X. (2024). Flow entry timeout hopping for SDN flow table LDoS attack defense. In *Intl. Conf. on Internet of Things and Cloud Computing Technology*.
- Humayun, U., Hamdan, M., and Marsono, M. N. (2021). Early flow table eviction impact on delay and throughput in software-defined networks. In *IEEE ICCSCE*.
- Isyaku, B. et al. (2020). Adaptive and hybrid idle–hard timeout allocation and flow eviction mechanism considering traffic characteristics. *Electronics*, 9(11):1983.
- Jain, L., Venkanna U., and Vollala, S. (2025). FTSheild: An intelligent framework for LOFT attack detection and mitigation with programmable data plane. *Expert Syst. Appl.*, 265:125865.
- Open Networking Foundation (2015). OpenFlow switch specification, version 1.5.1. Technical report, ONF.
- Pascoal, T. A., Fonseca, I. E., and Nigam, V. (2020). Slow denial-of-service attacks on software defined networks. *Computer Networks*, 173:107223.
- Tang, D. et al. (2024). FTODefender: An efficient flow table overflow attacks defending system in SDN. *Expert Syst. Appl.*, 237:121460.
- Wang, J. et al. (2025). FTOA-RP: A ‘group’-based flow entry replacement policy probing and flow table overflow attack method. *Comput. Secur.*, 159:104629.
- Yu, M., Xie, T., He, T., McDaniel, P., and Burke, Q. K. (2021). Flow table security in SDN: Adversarial reconnaissance and intelligent attacks. *IEEE/ACM Trans. Netw.*, 29(6):2793–2806.
- Zhao, Y. et al. (2023). P4LRU: Towards an LRU cache entirely in programmable data plane. In *ACM SIGCOMM*.