Título do Trabalho: Trabalho 2: Classificação de Tráfego por Volume e Encaminhamento Diferenciado com P4
Disciplina: Redes de Computadores
Integrantes: Gabriel Stiegemeier, Lucas Both Steinmetz Ribeiro, Mathias Eckert Recktenvald, Murilo Hesse Block
Professor: Carlos Raniery Paula Dos Santos
Data: 22 de outubro de 2025

1. Introdução

Este relatório descreve a implementação de um sistema de QoS em P4 que classifica fluxos UDP por volume de tráfego e realiza roteamento diferenciado via DSCP. Tráfego bem-comportado é encaminhado por canal de 20 Mbps, enquanto tráfego excessivo usa canal de 3 Mbps, com mecanismo de histerese para recuperação gradual.

2. Implementação da Solução

2.1 Topologia da Rede

A topologia consiste em dois hosts (h1: 10.0.1.1 e h2: 10.0.2.2) conectados via switches s1 e s2 com dois links paralelos redundantes. O primeiro link (portas 2-2) opera a 20 Mbps como canal de alta prioridade para tráfego bem-comportado. O segundo link (portas 3-3) opera a 3 Mbps como canal de baixa prioridade para tráfego que excede limiares, permitindo roteamento diferenciado baseado no comportamento dos fluxos.

2.2 Parâmetros de Classificação

O sistema usa janelas de 100 ms (131 ms reais via deslocamento de 17 bits no timestamp) com limiar de 100.000 bytes/janela (~8 Mbps). Fluxos acima desse valor são marcados como RED e direcionados ao canal de baixa prioridade. Para recuperação, implementa-se histerese com limiar de 12.500 bytes/janela (~1 Mbps): fluxos RED devem permanecer abaixo desse valor por 100 janelas consecutivas para retornar a GREEN, suportando até 8.192 fluxos simultâneos.

3 Modificações no P4

3.1 Cabeçalhos e Parser

Adicionado cabeçalho UDP para identificação de fluxos via 5-tupla. O parser extrai sequencialmente Ethernet, IPv4 (EtherType 0x800) e UDP (protocol 17), descartando outros protocolos. Metadados incluem: flow_hash (identificador), flow_channel (GREEN=0/RED=1) e distant_dst (flag de roteamento por DSCP).

3.2 Rastreamento e Classificação

Quatro registradores mantêm estado persistente por fluxo: flow_byte_count acumula bytes transmitidos na janela atual, flow_bucket armazena o identificador da última janela temporal observada, flow_channels mantém o estado de classificação (GREEN=0 ou RED=1), e flow_lastcrosses registra o bucket da última transição ou violação do limiar de recuperação. Cada fluxo é identificado unicamente via hash CRC32 aplicado à 5-tupla (IPs origem/destino, portas UDP origem/destino, protocolo), com resultado reduzido por módulo 8192 para indexação.

A ação measure_classify é o núcleo do sistema. Inicialmente lê o estado completo do fluxo e calcula o bucket temporal atual via timestamp global. Ao detectar nova janela temporal (bucket diferente), avalia a taxa de bytes da janela anterior: fluxos GREEN excedendo 100k bytes são imediatamente penalizados para RED, registrando o bucket da transição.

A lógica de histerese para recuperação opera da seguinte forma: se a taxa da janela anterior ainda exceder 12.5k bytes, flow_lastcrosses é resetado (violação). Se estiver abaixo, o sistema calcula quantas janelas se passaram desde a última violação - apenas após 100 ou mais janelas consecutivas com bom comportamento o fluxo retorna a GREEN. Qualquer taxa acima do limiar reseta o contador, impedindo oscilações rápidas. Após avaliações, os registradores são atualizados: bucket atual é salvo, novo estado escrito, e

bytes reiniciados com o tamanho do pacote atual. Na mesma janela, apenas soma-se o tamanho ao acumulado.

3.3 Marcação do DSCP e Roteamento

A tabela traffic_classification marca GREEN como DSCP 34 (AF41) e RED como DSCP 0 (Best Effort). A ação ipv4_forward retorna porta 0 para destinos remotos, ativando a flag distant_dst. A tabela dscp_routing então mapeia DiffServ=136 (DSCP 34) para porta 2 (20 Mbps) e DiffServ=0 para porta 3 (3 Mbps), separando tráfego por prioridade.

3.6 Processamento

O bloco apply executa: valida IPv4, verifica UDP, calcula hash, realiza medição/classificação, marca DSCP, consulta tabela LPM e aplica roteamento por DSCP se distant_dst=1.

4 Configuração das Tabelas

Ambos switches configuram traffic_classification identicamente (flow_channel 0→DSCP 34, 1→DSCP 0) e dscp_routing (DiffServ 136→porta 2, 0→porta 3). Em s1, ipv4_lpm roteia 10.0.1.1 para porta 1 (local) e 10.0.2.2 para porta 0 (remoto). Em s2, a configuração é simétrica, garantindo roteamento bidirecional consistente.

5 Validação e Resultados

Para validar a correta implementação da lógica P4 e das regras de controle, foram realizados quatro testes, um teste tentando estabelecer uma conexão não UDP e falhando e outros três testes simulando volumes diferentes de tráfego.

Para a simulação do tráfego, foram criados três arquivos em python que fazem o envio dos pacotes, sendo eles: *low_rate.py*, *high_rate.py*, e *histerese.py*. Eles são executados com o seguinte comando: *h1 python [arquivo.py] [ipdestino(h2)] [porta].*

5.1 Comandos de execução

Os testes foram conduzidos utilizando os seguintes comandos nos terminais do mininet.
Teste não UDP:

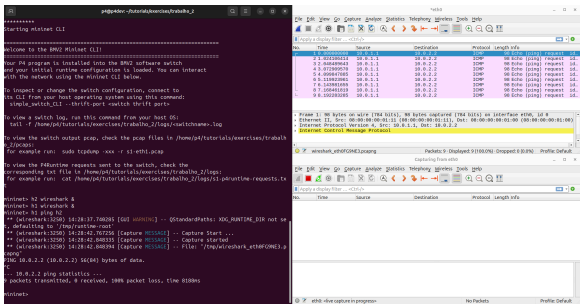
- h1 ping h2

Testes UDP com vazão variada:

- h1 python low_rate.py h2 5001
- h1 python high_rate.py h2 5001
- h1 python histerese.py h2 5001

5.2 Teste 1: bloqueio de tráfego não UDP

O primeiro teste visou confirmar o descarte de pacotes não UDP, através de um ping enviado de h1 para h2. O teste foi um sucesso e confirmou que h2 não recebeu nenhum pacote. Na janela da esquerda está o mininet; na janela da direita, o wireshark aberto em h1 no canto superior, e em h2 no canto inferior.

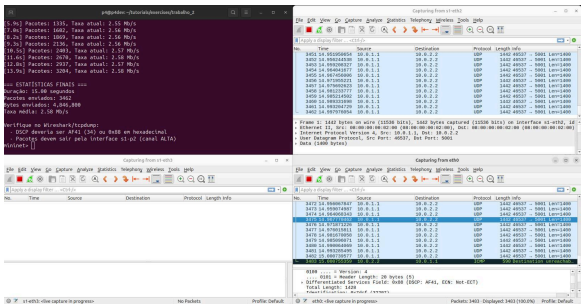


5.3 Teste 2: alta prioridade em tráfego de baixo volume

O próximo teste foi para confirmar que pacotes em baixo volume ficariam marcados com alta prioridade, DSCP 34. Para isso foi simulado utilizando python um volume de dados por volta de 2.5Mb/s enviados de h1 com destino a h2 pela porta 5001.

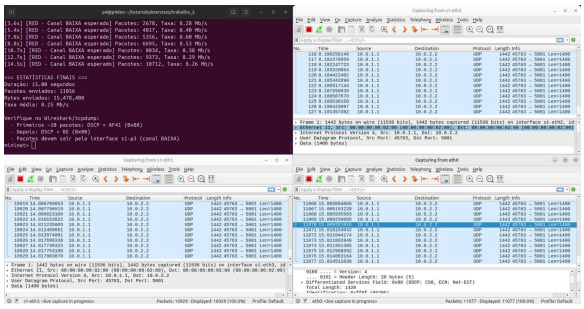
Nesse teste, e também nos subsequentes, os prints seguem a seguinte organização: no canto superior esquerdo está o mininet, no inferior, o wireshark no canal 3 do s1, destinado a baixa prioridade. No canto superior direito, o canal 2 do s1, destinado a alta prioridade, e no inferior está aberto o h2.

A imagem evidencia que h2 recebeu os pacotes de h1 marcados com alta prioridade. Sendo assim, o sistema funcionou de acordo com o esperado.



5.4 Teste 3: baixa prioridade em tráfego de alto volume

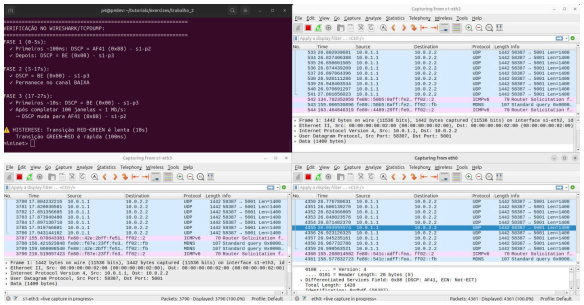
Nesse teste, o objetivo era verificar se um grande volume de dados de h1 em direção a h2 seriam marcados com baixa prioridade e encaminhados pelo canal de baixa. Para isso, foi feita simulação em python, com uma taxa média por volta de 8.2Mb/s enviado de h1 para h2 pela porta 5001. Dado o teste, foi percebido que os primeiros pacotes enviados por h1 passaram com alta prioridade, mas assim que foi feita a primeira verificação e se percebeu que a taxa de envio estava acima do limiar, os próximos pacotes foram enviados com baixa prioridade, sendo assim, o teste foi um sucesso.



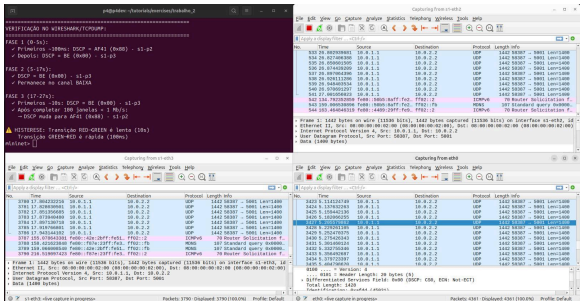
5.5 Teste 4: simulação de um tráfego variável para verificação do histerese

No último teste, foi enviado um tráfego variável de h1 em direção a h2. O tráfego é simulado pelo python no mininet pelo comando: h1 python histerese.py h2 5001, e é dividido em 3 fases. Na primeira fase é enviado um alto volume, os primeiros pacotes são marcados com alta prioridade e assim que percebe-se o volume eles são enviados com baixa prioridade pelo canal de baixa. Na segunda fase mantém-se o alto volume e os pacotes continuam sendo enviados com baixa prioridade. Já na terceira fase, diminui-se a taxa de transmissão, sendo assim, após 100 janelas seguidas com baixo volume (<1Mb/s), os pacotes voltam a ser enviados com alta prioridade pelo canal de alta.

O print a seguir mostra um pacote enviado próximo ao fim da execução do tráfego que chegou com alta prioridade.



Já esse print mostra um pacote enviado pela metade do tráfego, que chegou em h2 com baixa prioridade.



6 Conclusão

A implementação atende todos os requisitos especificados: classificação de tráfego UDP baseada em volume com limiar de 8 Mbps, marcação DSCP dinâmica, roteamento diferenciado por caminhos de 20 Mbps e 3 Mbps, e descarte de tráfego não-UDP. O mecanismo de histerese garante estabilidade ao sistema, exigindo que fluxos penalizados demonstrem comportamento adequado por 100 janelas consecutivas antes de recuperar alta prioridade. O sistema opera completamente no plano de dados, mantendo estado através de registradores e realizando decisões em velocidade de linha.