

P4 (Programming Protocol-independent Packet Processors)

Diego Nogueira Rezende Vilela GEC 1988

O que é P4 e seu propósito principal?

P4 é uma linguagem de programação de domínio específico voltada para descrever como dispositivos de rede processam pacotes no plano de dados. Seu objetivo é tornar switches, roteadores e NICs independentes de protocolos fixos, permitindo que engenheiros definam novos cabeçalhos, parsers e ações sem depender de hardware proprietário.

A linguagem nasceu como extensão das Redes Definidas por Software (SDN), ampliando a ideia de separar controle e encaminhamento. Enquanto o SDN clássico usa protocolos como OpenFlow para instalar regras em tabelas pré-definidas, o P4 permite programar todo o pipeline de processamento. Em resumo, ele oferece flexibilidade total no plano de dados, acelerando a adoção de novos protocolos e serviços de rede.

Principais características técnicas

P4 apresenta várias características-chave que a diferenciam:

- **Independência de protocolo:** o programador define completamente os formatos de cabeçalho a serem analisados. Não há limite fixo de protocolos; tanto padrões bem conhecidos (Ethernet, IPv4, TCP etc.) quanto protocolos personalizados podem ser declarados. Por exemplo, em P4 um cabeçalho Ethernet pode ser definido como abaixo, indicando os campos de origem, destino e tipo Ethernet (adaptado de exemplos da literatura):

```
1  header ethernet_t {  
2      bit<48> dst_addr;  
3      bit<48> src_addr;  
4      bit<16> ethertype;  
5  }  
6
```

Nesta declaração (exemplo adaptado), cada campo tem tamanho fixo em bits. Pode-se criar quantos *headers* personalizados forem necessários. Segundo Bosshart et al., “Switches should not be tied to any specific network protocols” – ou seja, o P4 permite experimentar novos cabeçalhos de rede sem mudar o hardware. Tudo deve ser explicitamente definido no programa P4, garantindo flexibilidade máxima.

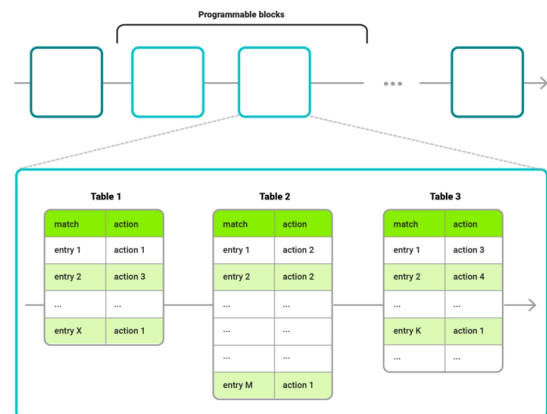
- **Programabilidade do plano de dados (match-action):** um programa P4 descreve toda a **sequência de processamento de pacotes** (pipeline). Em geral, o fluxo básico envolve um *parser* que extrai cabeçalhos do pacote, um ou mais blocos de controle (ingress/egress) compostos de tabelas *match-action*, e um *deparser* que

reconstrói o pacote para envio. Cada *tabela de correspondência* (match) é definida pelo P4, especificando campos-chaves (e.g. endereço de destino IP) e ações associadas (e.g. encaminhar, dropar, modificar campos). Por exemplo, um trecho P4 que define uma tabela de roteamento simples pode ser escrito como:

```
1 table routing {
2     key = { hdr.ipv4.dstAddr : lpm; }
3     actions = { drop; route; }
4     size = 2048;
5 }
6
```

Nesta tabela, cada entrada mapeia um prefixo de IPv4 (campo `hdr.ipv4.dstAddr`) a uma ação (`drop` ou `route`). O programador P4 também define as ações (funções) executadas. Todo o pipeline – desde como os cabeçalhos são analisados até que ações são aplicadas – é codificado na linguagem. Em P4, cabe ao desenvolvedor projetar esse pipeline de tabelas e fluxos de controle. Essa flexibilidade torna P4 uma solução poderosa para especificar lógicas de encaminhamento arbitrárias.

Figura: Exemplo de pipeline P4 com três tabelas match-action sequenciais. Cada tabela realiza comparações (*match*) de campos de cabeçalho e executa ações sobre os pacotes. As setas indicam o fluxo do pacote através dos blocos programáveis.



codilime

- **Independência de dispositivo (“target independence”):** P4 não assume um tipo específico de hardware. Um mesmo código P4 pode ser compilado para diversos alvos: ASICs programáveis (como Intel Tofino, da Barefoot/Intel), FPGAs, NICs inteligentes (P4 PNA – Portable NIC Architecture) ou até mesmo softwares de switch como o *bmv2*. Segundo Bosshart et al., um dos objetivos do P4 é que “os programas devem ser escritos independentemente dos detalhes do hardware subjacente”. Na prática, cada fabricante define um modelo de arquitetura P4 para seu dispositivo, descrevendo quais blocos programáveis existem no pipeline. O compilador P4 usa esse modelo para gerar as configurações adequadas para o hardware alvo. Ainda que o código deva respeitar o modelo-alvo (não é totalmente portátil entre arquiteturas diferentes), ele é abstraído do nível de portas físicas ou lógica fixa embutida. Isso promove portabilidade e evita o vendor lock-in: um programa P4

escrito para um modelo PSA rodaria em qualquer switch compatível com esse modelo, independentemente do fabricante.

- **Outras características:** P4 é estaticamente tipada e evita construções redundantes (por exemplo, comandos como `emit(hdr)` para deparsing são pré-definidos). O P4–16 (versão atual) introduziu um modelo de *package* e reforçou a separação entre definição de headers e o fluxo de controle. Também suporta estados simples (*counters*, *registers*, *meter*) via *externs*, mas esses detalhes podem variar por arquitetura. Em todo caso, a natureza de “pipeline” de P4 (sequência de tabelas match-action) permanece central.

P4 e Redes Definidas por Software (SDN)

O P4 é intimamente relacionado ao conceito de SDN, mas atua principalmente no plano de dados. No modelo SDN tradicional, um controlador centralizado gerencia o plano de controle, enquanto switches de forwarding (de hardware fixo) simplesmente executam regras instaladas (via protocolos como OpenFlow).

O P4 amplia essa visão: ele permite que o próprio comportamento do plano de dados seja programável e aberto. Em outras palavras, em vez de apenas instalar regras estáticas em tabelas fixas, o controlador P4 pode definir como o *próprio pipeline* deve ser estruturado. De fato, Bosshart et al. notam que o P4 foi concebido para funcionar “em conjunto com protocolos de controle SDN como OpenFlow”. Entretanto, ao contrário do OpenFlow clássico (que lista explicitamente campos fixos de cabeçalho), o P4 exige que o programador especifique o parse e o processamento inteiro dos pacotes. Isso eleva o nível de abstração: P4 “*configura um switch, dizendo como os pacotes devem ser processados*” e “aumenta o nível de abstração” no controle da rede. Em outras palavras, P4 pode ser visto como uma linguagem complementar ao controle SDN, onde o controlador define (em tempo de configuração) o comportamento do plano de dados via APIs como o P4Runtime. Essa API padrão e aberta permite que controladores (e.g. ONOS, OpenDaylight) manipulem dinamicamente tabelas e metadados dos dispositivos programados em P4. Assim, enquanto o SDN original tinha o switch como uma “caixa preta que segue instruções”, o P4 transforma o switch em um dispositivo *programável*, cuja lógica interna é definida pela rede, mantendo o controlador na função de orquestrar políticas de alto nível.

Aplicações práticas de P4 em redes

Graças à sua versatilidade, o P4 tem sido empregado em diversos cenários de rede reais. A seguir, alguns exemplos de aplicações práticas:

- **Redes de Telecomunicações (5G):** P4 está sendo usado para implementar funções do plano de dados de redes móveis modernas. Por exemplo, o núcleo 5G (core) conta com a *User Plane Function* (UPF), que atua como roteador IP, aplica QoS, mobilidade e outras funções de usuário final. Pesquisadores demonstraram que o UPF pode ser totalmente especificado em P4. O programa P4 define como o switch deve tratar mensagens de controle (PFCP) e encaminhar pacotes de usuário. Essa abordagem permite prototipar rapidamente novos recursos 5G e integrações com o control plane

aberto, já que o P4 pode gerar automaticamente interface de controle adequada. Projetos acadêmicos e industriais mostram implementações de UPF e gateways 5G baseados em P4, tirando proveito da flexibilidade para evoluir o padrão sem trocar hardware.

- Data centers e redes corporativas: Em ambientes de data center, P4 facilita soluções customizadas de balanceamento de carga, *multi-path* e telemetria embutida. Por exemplo, a telemetria em banda (In-band Network Telemetry – INT) pode ser implementada em P4, inserindo métricas de rede nos próprios pacotes à medida que atravessam o switch. O P4 permite também suporte a encapsulamentos avançados (como VXLAN, NVGRE, MPLS) sem depender do firmware fixo do switch. Em resumo, engenheiros podem definir políticas específicas de encaminhamento e transformação de pacotes para sua aplicação – desde marcação de pacotes a níveis de serviço até compressão ou enfileiramento customizados – diretamente no pipeline P4.
- Segurança e NFV: P4 tem sido usado para criar funções de rede de alta performance, antes implementadas em middleboxes. Firewall, detecção de intrusão (IDS/IPS), mitigação de ataques DDoS e NAT são exemplos de serviços que podem ser codificados em P4. Como P4 permite definir ações arbitrárias sobre pacotes, é possível inspecionar e modificar cabeçalhos em velocidade de linha (hardware), reagindo a padrões de tráfego suspeitos ou gerando métricas de segurança. Dessa forma, operadoras podem virtualizar funções de segurança (NFV) no próprio switch, obtendo ganho de desempenho e flexibilidade.
- Internet das Coisas e Infraestruturas Críticas: Em redes de IoT e sistemas legados (p.ex. automação industrial, SCADA), frequentemente há protocolos proprietários ou antigos que precisam coexistir com redes IP modernas. P4 pode atuar como *gateway* que traduz entre esses protocolos. Por exemplo, operações de rede elétrica (ICS) podem usar P4 para encapsular protocolos legados em IP e aplicar QoS específico, integrando sensores antigos a arquiteturas em nuvem. A capacidade de programar o parse de novos protocolos permite que dispositivos baseados em P4 façam conversões e filtrações que seriam inflexíveis em hardware tradicional.
- Virtualização e redes experimentais: Muitos laboratórios acadêmicos e empresas usam P4 para investigar novas arquiteturas de rede. Por exemplo, o Google tem trabalhado em projetos de SDN baseados em P4 (como o switch ToR reconfigurável “P4-programmable”), e provedores de nuvem (Amazon, Microsoft, Google) mostram interesse em usar P4 para personalizar roteamento interno. Além disso, P4 é usado em ambientes de emulação (Mininet/P4) para testar protocolos antes da implantação real.

Em suma, P4 está sendo adotado em roteadores de alta velocidade (Barefoot/Intel Tofino, Cavium/Marvell) e NICs programáveis, assim como em switches de software (BMv2). O fato de grandes empresas de tecnologia (Cisco, Broadcom, Alibaba, etc.) participarem do consórcio P4 indica a confiança na tecnologia. Como descrito em artigos recentes, hoje P4 é considerado “o *passo seguinte na evolução do SDN*”, abrindo caminho para redes definidas inteiramente por software, do centro de dados até as redes de operadoras.