

Frenetic: A Network Programming Language

Com o avanço da programabilidade de redes tem vindo a surgir várias linguagens de programação para este objetivo. No entanto estas linguagens, usadas para programar redes, têm a falta de características modernas, isto advém de serem linguagens de baixo nível de abstração, tendendo a serem bastantes rudimentares não dando suporte à programação modular, e os seus programas tendem a ser complicados, propensos a erros e difíceis de manter.

Devido aos problemas apresentados acima os autores desenvolveram uma nova linguagem de programação de redes, chamada Frenetic. Esta linguagem está embebida no Python e apresenta um alto nível de abstração. Pode ser descrita como uma linguagem de *queries* que facilita a classificação e a agregação do tráfego de redes e uma biblioteca para descrever políticas de pacotes de alto nível.

Esta linguagem de programação é composta por dois níveis de abstração, (1) um conjunto de operadores para construir e manipular o tráfego da rede e (2) um sistema em *run-time* que está encarregue de instalar e desinstalar regras nos *switches*.

A linguagem de *queries* apresentada permite aos programas lerem o estado da rede. De forma a implementar estas leituras eficientemente o Frenetic faz, em *run-time*, modificações ao estado da rede, instalando regras nos *switches*. No entanto, do ponto de vista do programador estas leituras e implementações não têm um efeito observável no estado real da rede. Um dos desafios da construção destas *queries* é encontrar o balanceamento entre a expressividade, simplicidade e controlo sob o custo. Tal como em todas as linguagens de programação, o desempenho é importante, muito mais quando se trata da rede. Por este motivo os programadores de Frenetic tiveram de perceber e saber utilizar bem os operadores básicos da linguagem de forma a obter o melhor desempenho possível, ou seja, perceber quando é que os pacotes devem ser desviados do seu caminho para o controlador, devido a uma consulta da rede.

A biblioteca de controlo de políticas permite aos programadores a instalação de políticas sobre a rede, esta instalação é feita através de eventos de políticas. Esta biblioteca vem com várias funções para ajudar os programadores a controlarem a rede, podendo estes criarem várias políticas de regras e instalarem nos *switches*.

O Frenetic foi construído a pensar nos programadores, para que estes não tenham de lidar com aspetos de baixo nível. No entanto os criadores do Frenetic tiveram que se preocupar com esse aspeto, traduzir o alto nível para o baixo nível. A arquitetura está dividida em três componentes: a própria linguagem Frenetic, o *run-time* e o controlador NOX. A utilização do NOX não é obrigatória, pode ser usado qualquer outro controlador no *backend*. O componente principal é o *run-time* pois esta é que se situa entre a linguagem Frenetic e o NOX. O *run-time* gere todo o trabalho em relação à instalação e desinstalação das regras nos *switches* e também estabelece os caminhos necessários para a comunicação entre estes e o controlador. Para traduzir a política de encaminhamento registada no *run-time*, é usada uma estratégia que reage aos fluxos à medida que estes ocorrem. No futuro os autores pretendem melhorar o *run-time* consoante as capacidades dos *switches* na rede.

De forma a testar o design do Frenetic os autores implementaram vários pequenos programas escritos usando a linguagem Frenetic e compararam-nos com programas equivalentes em NOX. Testaram então as linhas de código usadas, o tráfego do controlador e o tráfego total.

Nos seus resultados concluíram que o Frenetic consegue manter o mesmo desempenho que o NOX em termos de tráfego na rede, mas que são necessárias muito menos linhas de código para escrever o programa.